AFRL-HE-WP-TR-2005-0149

STINFO COPY

AIR FORCE RESEARCH LABORATORY

Work-Centered Technology Development (WTD)

Randall Whitaker

Northrop Grumman Information Technology
2555 University Blvd.
Fairborn, OH 45324

Ronald Scott

BBN Technologies LLC
10 Moulton Street
Cambridge, MA 02138

Emilie Roth

Roth Cognitive Engineering
89 Rawson Road
Brookline, MA 02445

Laura G. Militello
Laurie L. Quill

University of Dayton Research Institute
300 College Park
Dayton, OH 45469

Mona T. Stilson
Jeffrey L. Wampler
Terry W. Stanard
Gina F. Thomas-Meyers
Vincent A. Schmidt

Air Force Research Laboratory

March 2005

Final Report for July 2003 to March 2005

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2005-0149

This technical report has been reviewed and is approved for publication.

FOR THE DIRECTOR

//SIGNED//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
March 2005

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
July 2003 - March 2005

4. TITLE AND SUBTITLE
Work-Centered Technology Development (WTD)

5a. CONTRACT NUMBER
F33615-99-D-6001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62202F

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1710D117

6. AUTHOR(S)
[1]Randall Whitaker, [2]Ronald Scott, [3]Emilie Roth, [4]Laura G. Militello, [4]Laurie L. Quill, [5]Mona T. Stilson, [5]Jeffrey L. Wampler, [5]Terry W. Stanard, [5]Gina F. Thomas-Meyers, [5]Vincent A. Schmidt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
[1]Northrop Grumman Information Technology
2555 University Blvd.
Fairborn, OH 45324

[2]BBN Technologies LLC
10 Moulton Street
Cambridge, MA 02138

[3]Roth Cognitive Engineering
89 Rawson Road
Brookline, MA 02445

[4]University of Dayton Research Institute
300 College Park
Dayton, OH 45469

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
[5]Air Force Research Laboratory, Human Effectiveness Directorate
Warfighter Interface Division
Air Force Materiel Command
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TR-2005-0149

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The Work-Centered Support System (WCSS) is a stand-alone interface client that differs fundamentally from traditional graphical user interfaces (GUIs). A WCSS is use focused; it provides multiple forms of user work aiding within a unified cognitive framework. To date, WCSS research has been primarily limited to single position implementations. An important concern is the ability of the current WCSS methodology to scale up to support broader application and reliable repeatability of results. Risk reduction research needs to be accomplished in this effort, to investigate the robustness of the current WCSS methodology and identify extensions to address the shortfalls. So far, the manipulation of user interface behavior by WCSS users has been limited to manipulating agent thresholds through a prescribed number of variables. This capability needs to be extended to allow the warfighter to dynamically change/add data sources and spawn agents to monitor, correlate, fuse, and present data from across the net (multiple data sources), as missions change. The feasibility of such a capability needs to be investigated. This report presents the results of a multi-pronged research effort which has investigated multiple Air Forces Research Laboratory, Human Effectiveness Directorate (AFRL-HECS)-designated issues believed to offer substantial promise for achieving these risk reduction research objectives. These issues include: evolvable systems, collaboration support, work-centered design (WCD) methodology development, user interface design patterns, and advanced information composition.

15. SUBJECT TERMS Work-Centered Support System (WCSS), Cognitive Work Analysis, User Interface Development Tools, Work-Centered Design (WCS), Collaboration, Data Fusion

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	208	1Lt Mona T. Stilson
					19b. TELEPHONE NUMBER *(include area code)* (937)255-3795

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

CHAPTER I.
Evolvable Systems

At the outset of the WIDE 6.2 project, the Statement of Work outlined a task addressing the subject of 'Agent Management'. During the course of the project, the team jointly reached a consensus that the purposes of WIDE would be better served if this topical focus were shifted. In particular, it was believed that the topic of managing software agent technologies in the service of dynamic systems adaptation over time would be more constructive. As a result, the thrust of the 'Agent Management' task was discussed and collaboratively focused towards the topic we chose to label 'Evolvable Systems'. The work conducted under the aegis of this topic was documented in a draft article submitted to the journal *Ergonomics*.

For the purposes of including the results of this work in the WIDE 6.2 final report, the draft manuscript of the *Ergonomics* article has been approved by the AFRL customers as a delivery format. Beginning on the next page, the article manuscript (lightly reformatted for inclusion in the final report) is given.

The section numbering used in the article manuscript has been preserved.

The references cited in the article manuscript are appended at the end of the manuscript text, and are not duplicated in the References section of the broader final report document.

Evolvable Work-Centered Support Systems: Creating Systems Users Can Adapt to Meet Changing Demands

RONALD SCOTT†, EMILIE ROTH‡*, STEPHEN DEUTSCH†, SAMUEL KUPER§, VINCENT SCHMIDT§, MONA STILSON§ and JEFFREY WAMPLER§

†BBN Technologies, Cambridge, MA, USA

‡Roth Cognitive Engineering, Brookline, MA, USA

§Air Force Research Laboratory, Wright-Patterson AFB, Ohio, USA.

*Corresponding author. Email: emroth@mindspring.com

Work-Centered Support Systems (WCSS) coordinate domain visualizations and intelligent software within an integrated work-oriented framework so as to effectively support decision-making, collaboration, work management and product development. A WCSS to support weather forecasting and monitoring in a military airlift organization was developed and fielded. Field observations were conducted both prior and subsequent to implementation. A striking finding was the constant changes that operations personnel faced that impacted cognitive work (changes in goals and priorities; changes in scale of operations; changes in team roles and structure; changes in information sources and systems). The changing workplace demands that we observed and the modifications to the WCSS made in response are presented as a case study. For today's fielded systems, making the changes that are responsive to users' changing requirements in a timely manner is seldom possible. In the research presented in this paper we describe our initial steps in developing an approach that will provide operations personnel with the capability to adapt their system to quickly meet their changing requirements—an *evolvable work-centered support system*.

Keywords: work-centered support systems; evolvable systems; end user development, command and control systems; weather forecasting

1. Introduction

Military command and control organizations operate in a dynamic work environment. Geo-political changes, organizational changes, and opportunities to exploit new information sources all drive a need for rapid changes in software support systems to keep pace with the changing character of work. Unfortunately, military command and control organizations often operate in an environment supported by inflexible systems. Even simple user change requests can take months to be satisfied. The life cycle of a change request, from prioritization and assignment, through development, test, evaluation, and certification to deployment can significantly lag behind the pace at which work demands shift. In this paper we argue for the need to develop new design philosophies and tools to better support the evolving nature of work, and suggest an approach that fulfils this requirement.

Over the past several years we have been developing work-centered support systems (WCSS) to aid mission planning and Command and Control in a military airlift service

organization (Scott, Roth, Deutsch, Malchiodi, Kazmierczak, Eggleston, Kuper and Whitaker, 2002; in press). WCSS are designed to provide comprehensive support for the multiple aspects of work (e.g., decision support, product development support, collaborative support, and work management support) within an integrated work-oriented framework (Eggleston, Young and Whitaker, 2000; Eggleston and Whitaker, 2002; Eggleston, 2003). Our first system, called Work-Centered Support System for Global Weather Management (WCSS-GWM) was developed to support weather forecasting and monitoring and is currently installed and in use in the airlift service organizations' operations center (Scott et al., 2002). More recently we have been working on expanding the scope of support to cover command and control of mission flights more broadly, focusing on the processes involved in monitoring for and responding to unexpected changes that arise during execution of mission flights (Wampler, et al., in press).

As part of the work-centered design process we had the opportunity to perform field observations and structured interviews with weather forecasting and command and control personnel over a span of four years. Field observations were conducted in the operations center both prior to development of our initial design concepts so as to ground the design in the field of practice, as well as after the initial system was deployed (toward the end of the second year), so as to insure that elements of work that were unanticipated and not well supported would be uncovered and addressed (Woods, 1998; Dekker and Woods, 2000).

One of the striking findings of our observations during that period was the constant change that operations personnel faced in the work environment. This included:

- changes in goals and priorities of the work and complexity of problems faced;
- changes in scale of operations (both in terms of number of personnel and number of missions supervised);
- changes in roles and team and organizational structure;
- changes in information sources and information systems provided to support work

While some of the changes could, in principle, have been anticipated, many of the changes were in response to larger forces outside of the organization that could not have been predicted. Existing information systems were not able to keep pace with these rapidly changing needs. As a consequence we observed a variety of 'make-shift' strategies and 'home-grown' artifacts emerge in an attempt to compensate for the inability of existing software tools to adapt to the continuously changing requirements.
Due to the research and development nature of our project, we were able to rapidly modify the WCSS-GWM, in response to the changing needs. However, the experience convinced us of the importance of developing software architectures that can more readily accommodate change.

In this paper we describe the changes we observed during the period of initial introduction of the WCSS-GWM and the kinds of modifications that were required to the WCSS-GWM in response to those changes. Our findings are presented as a case study to

illustrate the challenges confronted in designing a WCSS to support a constantly changing environment. The results point to the need for software systems that can evolve to adapt to the inevitable unanticipated changes that arise in the world. We coin the term '*Evolvable Work-Centered Support Systems*' to describe the adaptable systems we envision and point to some promising software directions for achieving that aim as well as outstanding issues to be confronted. These include: (1) Can users as well as developers be provided with the tools to rapidly adapt their systems to changing workplace demands? and (2) Can test and evaluation procedures be developed that adequately support the process for rapid change in software capabilities?

We begin by providing an introduction to work-centered support systems, the work-centered design process used to develop them (Section 2), and an overview of the WCSS-GWM system that was developed following this process (Section 3). Section 4 examines the range of operational changes that were observed over a four year time span and their implications for design of evolvable work-centered support systems. The results of two analyses are reported that motivate the need for evolvable work-centered support systems, and point to the kinds of capabilities that evolvable work-centered support systems need to display. One analysis examined change requests that were submitted to the WCSS-GWM software design team by the user community. The second analysis examined the kinds of work-arounds and informal artifacts developed by the user community to compensate for inability of existing software systems to accommodate operational changes. Section 5 explores software technologies that can provide the underpinnings for development of evolvable work-centered systems. The final section discusses evolvable work-centered systems in the context of other similar calls for systems that can more readily adapt to unanticipated change.

2. Elements of Work-Centered Design

The WCSS-GWM was developed as part of a program to develop and demonstrate work-centered support system concepts and methods. In this section we introduce the concept of a WCSS and the work-centered design process used to define requirements for, build and evaluate a WCSS.

Over the past several years Eggleston and his colleagues (Eggleston, Young and Whitaker, 2000; Eggleston and Whitaker, 2002; Scott et al., 2002; Eggleston, 2003; Eggleston, Roth and Scott, 2003; Scott, et al., in press) have developed WCSS design concepts and principles intended to support the multiple aspects of work as well as a framework for design and evaluation of WCSS. Elements of support considered in a WCSS approach include:

1) *Decision Support*: aiding problem solving and other cognitive processes in the process of performing work;
2) *Product Development Support:* aiding of the production of the deliverable artifact(s) of work;
3) *Collaborative Support*: aiding team and colleague interactions in work, and

4) *Work Management Support:* aiding the metacognitive activities entailed in prioritizing and managing the multiple interwoven tasks that normally arise in work.

A work-centered design framework has been developed that defines a design process for developing WCSS (Eggleston, 2003). The elements of the design framework are shown in Figure 1.

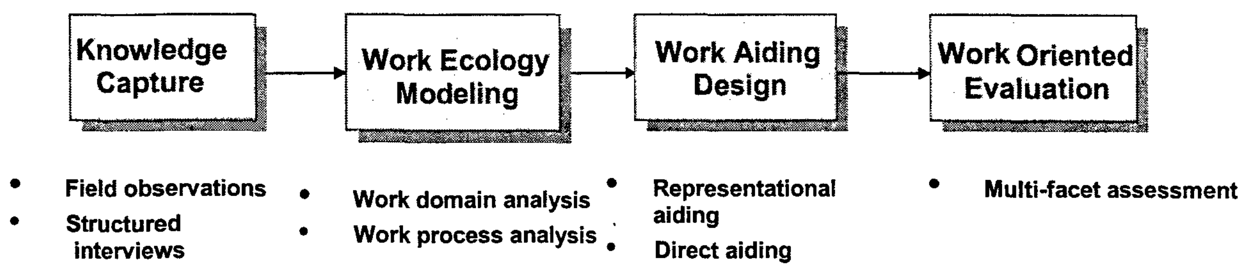


Figure 1. Overview of the Work-Centered Design (WCD) framework (adapted from Eggleston, 2003).

A fundamental aspect of the WCD approach is an analysis and modeling of the *work ecology* to uncover the elements of work that require support. The process starts with knowledge capture methods such as ethnographic field observations and structured interview techniques (e.g., Militello and Hutton, 1998; Roth and Patterson, 2005) to uncover the characteristics of the work domain, the work requirements, the sources of complexity and cognitive and collaborative demands entailed. Formal methods can then be employed to represent the results of the analysis (work ecology modeling). These include work domain analysis methods that model the intrinsic characteristics of the work to be achieved (Vicente, 1999; Elm et al., 2003) as well as methods that model workflow dynamics within and across individuals and groups required to achieve work goals (Kirwan & Ainsworth, 1992) and use scenarios that illustrate particular work threads requiring support (Carroll and Rosson, 1992).

The products of work ecology modeling define the work-aiding requirements to support domain practitioners in performing work in a flexible and adaptable manner, given the dynamics of the work context. These requirements are used to guide work-aid design that involves development and prototyping of work-aiding concepts. Work aiding may take the form of *representational aiding* (Woods and Roth, 1988) that is provided through the use of work domain visualizations or *direct aiding* provided by a coordinated set of software agents that interact with the user that are clearly connected to or are embedded in the work domain visualizations (Eggleston, 2003; Scott, et al., in press).

Work-aiding design involves a process of iterative refinement through multiple prototype development and user feedback cycles. The importance of incorporating user evaluations as part of the work-centered design process is highlighted by the last box in Figure 1 that explicitly calls out the need for multi-faceted empirical evaluation at multiple points in

the development cycle. Evaluation is focused on assessing the extent to which the proposed elements of support embodied in the prototype actually provide the envisioned support (Woods, 1998; Potter, Roth, Woods and Elm, 2000; Woods and Dekker, 2000).

3. Developing A WCSS for Assessing Weather Impacts on Airlift Missions

The WCSS-GWM was developed to support weather forecasting and monitoring in a military airlift service organization. It employed the work-centered design methodology and was intended to provide an illustration of a work-centered support system in a command and control organization.

Traditionally, airlift pilots have been responsible for their own flight planning, including obtaining pre-flight weather briefings. In this organization, a new approach was initiated to reduce the amount of time an aircrew had to devote to these tasks. A flight manager (FM) position was created with the primary responsibility for planning and managing multiple flights, both pre-flight and en route. This includes obtaining a weather briefing and providing a complete flight plan to the pilot, including weather forecast information. The FM is viewed as a 'virtual crew member' in support of the pilot. Weather can significantly influence pre-flight and en route flight management decisions (e.g., there may be a need to accelerate, delay or re-route a flight due to unfavorable weather conditions). As a result, weather forecasters must work closely with the FMs to evaluate weather conditions at the departure and arrival airfields as well as along the planned route. The focus of our effort was on developing an intelligent system to aid near-term weather forecasting in support of planning and managing airlifts, both pre-flight and en route.

At the time the study was initiated (February 2001), FM and weather forecasters worked closely to determine the potential impact of predicted weather on the viability of upcoming flights. If hazardous weather conditions were forecasted (e.g., high turbulence or lightning) then the FM and weather forecaster worked collaboratively to identify alternative routing that would avoid the problematic weather areas. However, they had limited software tools to support their collaborative decision-making processes. While the weather forecasters had various displays available for actual and predicted weather in different parts of the world, the information came from multiple sources and was presented on separate displays. Further, there were no graphics depicting the planned flight paths of upcoming missions making it necessary for forecasters and FMs to mentally fuse the various sources of disparate information to assess the potential impact of weather on a mission.

The WCSS-GWM was designed to address this problem. It combines integrated visualizations to enable the weather forecasters and FMs to directly 'see' the impact of weather on flight missions as well as intelligent software agents that monitor weather conditions and provide alerts when specific weather conditions may operationally impact current and planned missions. Here we provide a brief overview of the WCSS-GWM system. Scott et al. (in press) provides a more complete description of the system and the

human-centered system design philosophy it embodies. Eggleston et al. (2003) provides a description of a work-centered evaluation of the WCSS-GWM.

The WCSS-GWM combines computer generated flight plans and weather information on a geo-spatial display. Layer controls allow flight and weather information (e.g., PIREPS, ACARS; airfield and upper air forecasts and satellite images) to be overlaid or removed from the map. Important features of the WCSS-GWM are the software agents that monitor missions, forecast and watch areas, and provide notification when operationally significant changes in weather arise. A screenshot of the WCSS-GWM display depicting the ability to create and modify software agents is provided in Figure 2.

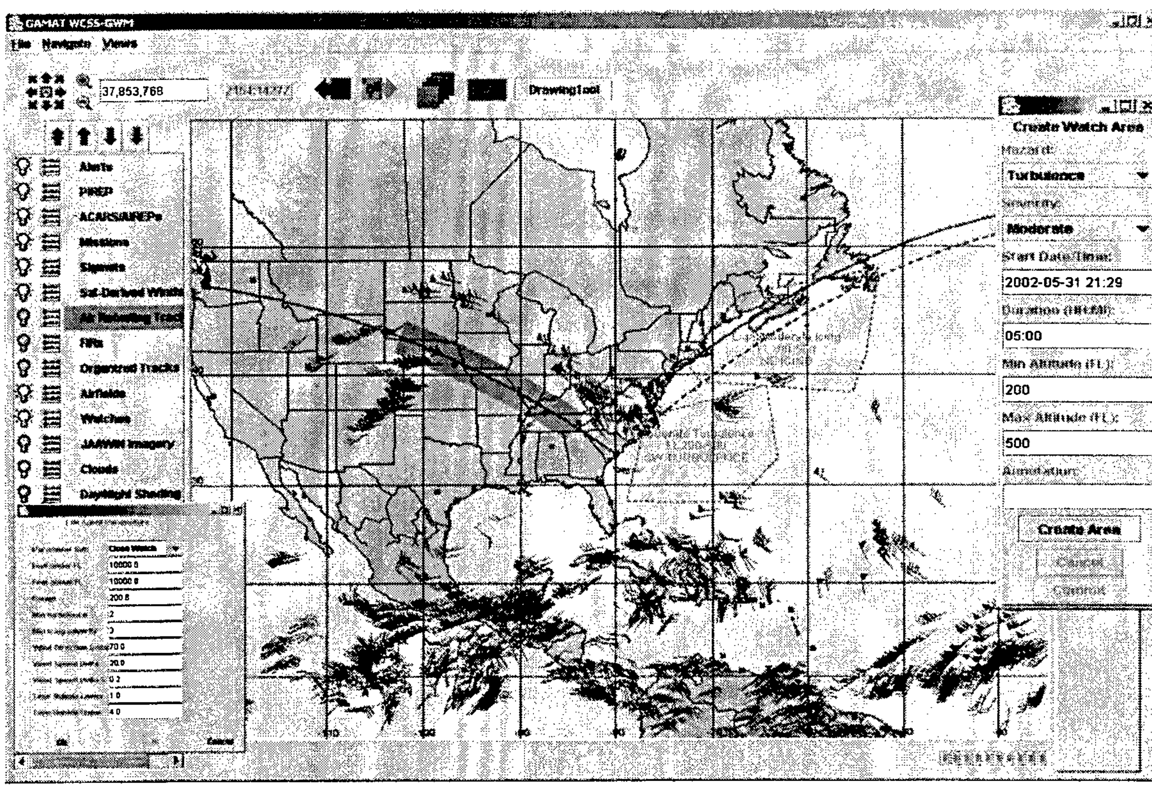


Figure 2. A screen shot from the WCSS-GWM that illustrates the ability to create and modify software agents.

The WCSS-GWM exemplifies and extends Cognitive Engineering principles for effecting human-software agent interaction and work-centered support system (WCSS) concepts. Consistent with a growing body of cognitive engineering literature (Roth, Malin & Schreckenghost, 1997; Christoffersen & Woods, in press), the software agents were explicitly designed to enable —*observability* and *directability by the user*. Users need to be able to 'see' what the automated agents are doing and understand what they will do next relative to the state of the task. They also need to be able to control and re-direct the software agents as task requirements change. The WCSS-GWM agent-based architecture was designed with these objectives in mind. The geo-spatial map with weather and flight information superimposed provides a "common ground"

representation of the current world state that is available to the humans (the FMs and the weather forecasters) and the software agents that are involved in interpreting weather-related information and its implications for flying missions. Furthermore, the activities of the agents are directly visible and controllable by the users—the geographic area being monitored by the software agents (both with regard to a mission route and with regard to forecast and watch areas) is explicitly presented on the display and can be modified by the user. Similarly, the weather-related parameters being monitored by the agents and the trigger points for alerts can be inspected and modified by the users.

4. A Constant Need to Respond to a Changing World

The WCSS-GWM development program covered three years from initial requirements gathering through handoff of a 24x7 operational system. The first year was largely devoted to initial understanding of the domain, the systems supporting the present day work flow, and exploration of the possible Work-Centered Support Systems that might be implemented. The WCSS-GWM system was designed, implemented and installed during the second year. While the initial functionality was not a complete solution (i.e. not as complete as dictated by the design process), weather forecasters and FMs began daily use of the system. Feedback from users guided the refinement of the system over the third year, as the system was completed.

A fourth year has elapsed during which time we have conducted additional observations and interviews in the command and control operations center as part of an ongoing program to expand the work-centered support for command and control staff. During the four year period we periodically conducted field observations in the operations center and structured interviews with command and control staff. These field visits occurred approximately every three months and were of two to three days duration.

The work environment of the airlift service organization did not remain static over the four year period of observation. Among the changes observed included:

- changes in goals and priorities of the work (e.g., the nature of flight missions that were conducted; the parts of the world where missions operated);
- changes in scale of operations;
- changes in roles, team and organizational structure;
- changes in complexity of problems faced (as number of missions increased the airlift service organization hit against hard resource constraints making it more important to anticipate and respond to resource bottlenecks and prioritize among missions in cases of goal conflict);
- changes in information sources and information systems provided to support work;
- and changes in the physical layout of the operations center (the operations center was remodeled with the result that forecasters and FM were no longer in as close physical proximity).

One of the most striking changes was in scale of operation. As work on the WCSS-GWM program was starting, in February 2001, the position of FM was just being created and staffed. The FMs were only assigned a small percentage of the flights handled by the Command and Control Operations Center of the Airlift Service Organization. Initially, there was an average of three FM per shift and FMs handled less than 20 flights a month. By February 2004 there was an average of 10 FM per mission and FMs handled more than 3000 flights a month.

With the increase in scale there was also a shift in team member roles and tasks. While initially forecasters worked one on one with a flight manager to produce a "tailored forecast" for each flight managed mission, the nature of the collaboration between forecaster and flight manager changed as the number of FMs and flight managed missions increased. The forecaster and FMs now needed support in identifying and managing a set of "high-risk" missions to focus on, treating those differently from the more routine missions. A series of system change requests were made to allow the WCSS-GWM to import information about "operational risk management" – identifying the high-risk missions and sorting and filtering missions based on risk assessment factors.

Technological changes occurred as well. One of the forecasters' primary responsibilities was preparing forecast hazard charts – maps that identified regions of forecast turbulence or icing hazards. It was originally envisioned that the forecaster would use the WCSS-GWM map tool to prepare these charts. By overlaying air mission flight plans on these forecast charts, a flight manager could see exactly which missions were likely to run into en route weather problems. A new weather forecasting software system came on-line for forecasters to use in preparing forecast hazard charts. While the new system provided much more detailed weather information, it had no capability to overlay flight plans on the same map as weather data. This led to a new requirement on the WCSS-GWM – the import of forecast chart data produced by the new system and the overlay of it on the WCSS-GWM map.

The WCSS-GWM was originally conceived of as a tool to aid the collaboration between weather forecaster and flight manager in identifying mission-endangering en route weather. As it came into daily use by both forecasters and flight managers, a number of system change requests were made in order to expand the utility of the system. Most of these changes involved bringing new data into the system and overlaying new information on the maps – air routes, Flight Information Region and country boundaries, and real-time position reports. All of these changes expanded the domain of the WCSS-GWM into areas of the flight manager's job that had not been the target application for the system as designed.

In some instances, a change request was made by the weather forecasters to support uses that were entirely unanticipated. At one point WCSS-GWM users began faxing maps to crewmembers. A request was made to alter some map symbology to make sure information could be correctly interpreted from a black-and-white fax copy.

Among the consequences of the changes we observed in the operating environment was a growing mismatch between the support provided by the information support systems in place, WCSS-GWM included, and the requirements of work. This led users to submit numerous software change requests. Because the WCSS-GWM development was an R&D effort, the software development team was in a position to rapidly respond to change requests. This is much different than the process for change in the current legacy systems. Even simple user change requests to legacy systems required lengthy lead times on the order of months to years to satisfy. The life cycle of a change request (from prioritization and assignment, through development, test, evaluation, and certification to deployment) significantly lagged behind the pace at which work demands shifted. As a consequence, we observed users turn to development of informal artifacts including 'home-grown' software to compensate for system – work mismatches.

Examination of user request changes and informal artifacts that emerged to compensate for rigid systems provided insight into the kinds of change mechanisms an evolvable work-centered system requires to support the evolving nature of work.

4.1 Analysis of WCSS-GWM Change Requests

To provide structure to our task, we looked at a very concrete example – the WCSS-GWM system – to understand what system changes have been requested, and the underlying reasons for those change requests. We have identified 50 system change requests. We classified these in two orthogonal ways – the underlying reason for the change request, and the impact on the supporting software to accomplish the change request.

By classifying the reason for the change request we hope to understand the source of our problem – how many of these change requests should have been anticipated? How many of these requests were based on changing requirements, as opposed to requirements we might have understood from the start? Could we have approached our requirements gathering and design work in a different way to eliminate the later need for some these change requests?

The goal of the exercise was to understand which change requests resulted from changes in the context of work that could not have been anticipated ahead of time and to provide a characterization of types of software changes they entailed. Understanding the kinds of software changes that are motivated by changes in the world can provide the basis for defining the kinds of mechanism for change that need to be provided in evolvable work-centered systems to enable users to adapt the systems to the changing nature of work.

Table 1 summarizes the classification of WCSS-GWM system change requests based on the reason for the request. It is clear that one of the most common reasons for a change request was expansion of the role of the WCSS-GWM within the organization – either its use by a new category of user, or by expanding the use by an existing user into a new area of work. It should be no surprise, based on the discussion of the previous section,

that another common reason for change was environmental change: some externally-triggered alteration in data availability, hardware, or software that induced new constraints on or offered new opportunities to the WCSS-GWM.

Table 1: Reasons for WCSS-GWM Change Requests

Reason for Change Request	Number of Change Requests	Comment
New user	11	Additional types of users resulted in expansion of the envisioned uses for the aid.
New use	6	Original type of user, but expanded scope of use.
Unanticipated model of use	3	Original type of user and scope of use (what they would use it for), but unanticipated model of use (e.g., when and how they would use it)
In queue	10	Anticipated functionality on the 'queue' of features to be eventually implemented, implemented as resources allowed
Environmental Change	10	Changes in hardware, software, data availability that impose new constraints or create new opportunities
Uncovery of Requirement	2	Uncovery of an existing requirement that was not picked up earlier (e.g., due to KA sampling limitation)
Change in work process	2	Change in the process by which work is conducted.
Organizational change	1	Change in the structure of the organization, change in how work is allocated across individuals and groups
Correction	1	Correction of a system problem
Design Improvement	3	Improvement of design, based on user feedback/testing
Organizational conflict	1	Reconciliation of disagreement between user organizations

Relatively few of the system change requests come about for reasons unrelated to changing user base or environment. These categories include uncovering existing requirements that had not previously been noted, correcting system problems, or even making modifications based on user feedback. The lesson to be learned from this table is that the bulk of system change requests, at least for this system, arise from changes in how the system is to be used, what other systems this one needs to communicate with, or other environmental changes surrounding this system. These are all changes that cannot be anticipated during the original design process.

The second classification of system change requests attacks a different side of the problem. We classified the change requests based on the type of software change it required. Is the change an addition of new information to an already existing display? Does it require an entirely new display to be developed? Does it require a new source of data to be integrated into the system? Is it a simple change in the rules identifying an alertable weather condition? By classifying system change requests in this way, we

hoped to gain an understanding of the kinds of software changes that are likely to be requested in future systems. This is a first step on the path to defining the kinds of changes that an evolvable system will need to be able to accommodate, and identifying design principles governing the development of evolvable decision support systems.

We classified change requests into four broad categories of software impacts – Data Acquisition changes, Automated Analysis changes, User Interface (UI) changes, and Software Infrastructure changes. The results are shown in Table 2.

The first category of software change is *Data Acquisition* software changes. Within this category the most common change requested was to begin to acquire a new type of data from a new source – new satellite cloud images, real-time position reports on en route missions, or new forecast hazard charts, for example. In a few cases, other changes need to be made to Data Acquisition software to accommodate data format changes or source changes.

Table 2: Software Impacts of WCSS-GWM Change Requests

Category of Software Change	Subcategory	Number of Changes
Data Acquisition		
	Acquire new data	9
	Change source/format for existing data	3
Automated Analysis		
	Add new analysis agent	6
	Add new processing module for use by an agent or GUI	6
	Modify rules of existing analysis agent	1
User Interface		
	Add new data to existing display	12
	Change how data is displayed	3
	New type of display	2
	New functionality	10
	Reorganization of GUI elements	4
Software Infrastructure		5
note: some changes require more than one category of software change		

The second most common software change resulted in *Automated Analysis* changes. In our terminology, Automated Analysis includes any automated processing (rule-based or otherwise) of information that assists in a decision about what information to display in the UI, how to prioritize information in a display, or how to display information. In the WCSS-GWM system, Automated Analysis rules are used to alert the user to missions scheduled to fly through forecast hazard areas. Automated Analysis rules are also used to color-code airfields, showing airfields operating under visual flight rules in green and airfields with worse flying conditions in a succession of other colors. One of the most common software changes related to Automated Analysis was to add a new rule-based

analysis agent to create a new type of alert. Other typical software changes involved creating new algorithmic procedures to be used by existing agents or GUI displays.

More than half of the system change requests involved changes to the *User Interface*. The two most commonly requested UI changes were adding a new type of data to an existing display (adding display of air routes or country boundaries, for example) and providing entirely new functionality in the UI. Newly requested UI functionality could be quite straightforward (adding new sorting and filtering capabilities) or could be quite complex (provide a new mechanism for re-ordering map layers). Less common UI changes involved defining entirely new display types, changing the way information is presented in an existing display, or even redesigning the organization of tool palettes.

The final category of software change, *Software Infrastructure* changes, generally resulted from change requests based on system environmental changes. These requests were initiated by new security requirements (replace FTP use by HTTPS) or new network configurations (interact with a newly-placed caching proxy server), for example.

Having classified the system software impacts of the change requests made of the WCSS-GWM, it should be clear that there are a number of these change requests that just cannot be handled by the user organization. Software Infrastructure changes, for example, may have to be handled the traditional way, with software engineers making the changes and delivering an updated product at a later time.

On the other hand, a significant number of system change requests resulted in simple UI changes (adding new data to an existing display) and/or straightforward Data Acquisition changes – adding a new data source. The impact of designing a Work-Centered Support System that could easily accommodate these changes by the end-user organization would be high. We estimate that more than half of the system change requests for the WCSS-GWM is of types that could be satisfied by the end-user organization operating an evolvable work-centered system.

4.2 User Strategies for Coping with a Changing World

Examination of change requests to the WCSS-GWM provided one window into the requirements for software system adaptation to keep pace with evolving work requirements. Examination of how the domain practitioners struggled to adapt existing software tools and created informal artifacts to compensate for limitations in those tools, provided a second window.

It has long been noted in the human factors literature that users will informally tailor their tools to more effectively meet the demands of the work domain (Vicente, 1999). Seminara, Gonzalez and Parson (1977) documented how power plant operators added labels to similar-looking displays and changed knobs on controls to make them easier to tell apart. More recently, Mumaw, Roth, Vicente & Burns (2000) and Vicente, Roth and Mumaw (2001) documented a number of ingenious strategies that operators developed to compensate for limitations in computer-based information and display systems and make

them better suited for support of the work. For example, operators were observed to modify alarm set points to create new alerts and reminders for action in situations not directly supported by the system as designed. Examination of the informal artifacts and strategies users develop can provide guidance for how to build WCSSs that more effectively support adaptation to evolving circumstances.

In this section we examine some of the informal artifacts we observed staff in the command and control operations center develop and use. The examples are drawn from observations of command and control staff responsible for detecting and addressing mission problems that arose during mission execution. Unlike the FM, they were not responsible for creating detailed flight routes, and did not use the WCSS-GWM system.

Over the course of our field observations we identified a number of cases where informal artifacts were created to compensate for the limitations and rigidity of existing information systems. These took the form of:

1. Physical artifacts such as handwritten cheat sheets and sticky notes;
2. New visualizations that graphically depicted important information that was not provided by the information systems as designed;
3. 'Local' databases that stored updates and corrections in information stored in the formal system data bases;
4. New software tools programmed by members of the user community to create support systems for aspects of work that were not well supported by the formal information systems.

Physical artifacts generally took the form of hand-written or typed 'cheat sheets' that provided summary reminders of factors that need to be considered in developing and modifying flight plans (e.g., the location and direction of legal air routes at different times of day). The use of informal physical artifacts is virtually universal across domains, and was thus not surprising to observe (Vicente, 1999).

More surprising was the emergence of locally developed software 'artifacts' such as new visualizations, local databases and 'home-grown' software tools that have not been as widely documented. They point to opportunities to provide more effective work-centered support by providing capabilities to more easily develop these local software 'artifacts' and link them to formally developed work-centered support systems.

A salient example of a new visualization was a case in which users modified an existing timeline display intended to support command and control staff in identifying situations where more planes were scheduled to land at a given airfield than could be accommodated. The display, as designed, focused on displaying the number of aircraft scheduled to land at an airfield as the primary indicator of the viability of current landing schedules. However, there were additional important factors the command and control staff needed to consider that were not visible in the display as designed. These were the operating hours of the airfield, which could change on short notice, and whether the scheduled landing time was during night or day since there could be restrictions on

whether planes could take off and land during those periods. The users came up with an ingenious way to graphically depict these important types of information on the airfield displays. They defined 'pseudo-planes' that did not actually exist and scheduled them to be at the airfield during the critical times in question (i.e., when the airfield was supposed to be closed; or when planes were not allowed to fly in or out). By entering these 'pseudo-planes' into the display system, they were able to create graphic visual indicators of information critical to their decision-making that was not anticipated as important in the original system design. This example is similar to examples observed by Vicente et al. (2001) of operators creating new indicators and alarms.

The command and control staff was also observed to create and maintain local databases that were more accurate and up-to-date than the information stored in the existing, formal, operational system databases. An example is a list of base operating hours and temporary closures. While the operational information system contained fields for base operating hours, the information was often out of date. Bases changed their operating hours and declared temporary base closures on short notice. The update cycle for the operational information system databases was not able to keep up with these changes. As a consequence the user community developed their own private, local, databases. One of the limitations of these private, local, databases, is that they are difficult to share, even among staff within the operating center. It was not unusual for multiple, individual controllers to each maintain their own private database - each containing slightly different information. One of the benefits of creating evolvable work-centered systems, with explicit provisions for the development and maintenance of local databases, would be the ability to make these local databases shared across multiple individuals fostering shared situation awareness and facilitating collaboration.

The most striking cases of software-based artifacts that we observed were instances where the user community developed their own software tools to support aspects of work that were not well supported by the formal software systems provided and maintained by the larger organization. We observed two clear examples, one developed by the weather forecasting staff and one developed by the command and control staff. In both cases the tools were built by a member of the user community using 'off-the-shelf' spreadsheet and word processing software. Macros were used to import data from the operational information systems, process and integrate it with locally available information, and then create new displays that better supported the work processes. In the case of the weather forecasting group, the large increase of missions to be monitored created a need to classify missions into different risk level categories based on a combination of weather related criteria. There were no provisions in the existing information systems for defining, displaying, or using these risk levels. Consequently, one of the forecasters developed a spreadsheet program to classify and manage missions by risk level.

In the case of the command and control staff, they needed a way to track more closely the subset of missions that were considered to be 'high visibility' or that had problems (e.g., missions delayed due to maintenance problems). They created a 'notepad' tool using standard word processing software with macros that allowed them to import information about these missions from the formal information systems, and add detailed annotations

as to the current status of the missions and planned actions. Macros allowed the notepads to be periodically updated so that the user could be alerted to new problems. These notepads served as a focused 'to do' list for the user enabling them to prioritize and manage their work, as well as a shift-turn-over log, allowing critical information to be shared across shifts supporting across-shift coordination and collaboration.

These various examples of 'home-grown' software-based artifacts provide salient examples of the creative work-arounds that users employ to compensate for mismatches between rigid software tools and the evolving demands of work. They point to the importance of developing systems that can be more readily modified by users to support their work.

5. Toward Evolvable Work-Centered Support Systems

The two previous sections have presented our observations on change mechanisms an evolvable work-centered support system would likely be asked to support. First we summarized the system change requests made of the WCSS-GWM. We followed with a description of some of the ways we have seen domain practitioners "work around" their current rigid systems to support new work requirements. These two sources of information served as raw material for deriving requirements for evolvable work centered support systems.

We propose a set of capabilities that an evolvable work-centered system should have, and discuss, at a high level, the software techniques that make these capabilities achievable. We acknowledge at the start that at present, as a field, we do not know how to build work-centered systems to easily support all the different types of modifications we envision. However, we can build systems that are more easily modifiable than traditionally-designed systems. And as we gain experience in designing evolvable work-centered systems, we will be able to better support evolvability. In this section we offer some first steps in that direction.

Section 5.1 describes the software structure of a prototypical work-centered support system, in order to provide a vocabulary to discuss evolvable work-centered systems. In Section 5.2 we list necessary capabilities for an evolvable work-centered system. Software architecture and technologies appropriate for implementation of evolvable work-centered systems are discussed in Section 5.3. Finally, in Section 5.4 we discuss some important issues concerning the deployment of evolvable work-centered systems in a command and control environment.

5.1 A Prototypical Work-Centered System

The goal in building an evolvable work-centered system is to provide functionality to the operating organization that will better support the changing nature of the work requirements and will be able to be modified much more quickly to track necessary

changes than traditionally-designed systems. To allow for changes to be made more quickly does not necessarily mean that the changes are to be made by the end user. Changes may be made by various roles within the operating organization, from developer to system administrator to user.

To bring concreteness to this discussion of what kinds of system changes an evolvable work-centered system can accommodate, we describe a prototypical work-centered system. Not every work-centered system will follow this model, but a work-centered system is likely to have enough in common with this prototype to make this discussion worthwhile.

Our prototypical work-centered system consists of three main components. First is the *Data Acquisition Module* - the component of the system that is responsible for acquisition, decoding, and storage of data in which any users may have an interest. Second is the *Analysis Module.* In the Analysis Module, typically some combination of rule-based and algorithmic code, the raw data acquired by the Data Acquisition Module is filtered and transformed into higher-quality information ("decision-quality information") of immediate interest to the user. For example, in the WCSS-GWM, the Analysis Module identifies particular upper-air turbulence observations that threaten the successful operation of air missions. Finally is the *Presentation Module*, which includes the Graphical User Interface (GUI) with which the user interacts directly as well as reasoning algorithms which try to prioritize information for viewing by the user.

The ideas we describe here about evolvable work-centered systems largely grow out of our work in developing the WCSS-GWM. While we cannot claim the WCSS-GWM to truly be an example of an evolvable work-centered system, we can describe the features of the WCSS-GWM architecture that accommodate certain types of changes.

The WCSS-GWM is structured as a client-server application. The server contains the Data Acquisition and Analysis Modules; the client contains the Presentation Module. While the server was originally intended to serve only four or five clients, the server in practice serves up to twenty clients. A later re-implementation concentrating on scalability requirements gave us a slightly modified architecture in which dozens of clients may be served.

Each of the server and client processes is implemented as a "scenario", or loosely coupled set of software agents, using the D-OMAR (Deutsch, 1998) distributed agent architecture. In the D-OMAR terminology, an agent is basically the manager of a small set of work-related threads of execution, which can interact with other agents through a publish-subscribe protocol. D-OMAR provides methods for agents to be initiated, to subscribe to signals, and to publish signals for other agents.

The Data Acquisition Module is made up of a number of independent agents – one agent for each data source (i.e., one agent to get cloud images from a National Oceanographic and Atmospheric Administration website, one agent to get information about the set of air missions flying that day, one agent to receive the latest tropical storm bulletins, etc.).

These data sources are listed in a configuration file read by the server as it starts up. As the configuration file is processed, a data acquisition agent is instantiated for each data source.

The Analysis Module is also made up of a set of independent agents. A single analysis agent is instantiated for each air mission to be monitored. This agent is responsible for watching the area in front of the mission for any indication of mission-threatening weather conditions. If the agent finds any such weather, it will create an alert, which is passed to the Presentation Module for display to the user.

The Presentation Module runs on the WCSS-GWM client. The agents of the client receive data to display from the WCSS-GWM server and prioritize missions and alerts for display to the user.

While the WCSS-GWM was not developed with evolvability in mind, we can claim a certain degree of evolvability for it. One example in particular shows the effectiveness of this architecture of loosely coupled software agents as a basis for building evolvable systems.

The WCSS-GWM had been designed to receive a feed of composite world-wide satellite images. A single agent in the Data Acquisition module received these images; the client displayed them as an overlay on the WCSS-GWM map. This worked well until one day the organization that provided the composite world-wide satellite image unexpectedly decided to no longer make such images available. The best replacement that could be found was a set of five satellite images that together covered most of the world. By editing configuration files, without changing any compiled software, the WCSS-GWM was reconfigured to accept this change. Instead of a single data acquisition agent, there were now five, each receiving one of the new satellite images. Instead of a single GUI control to turn on and off the satellite image on the client, there were now five GUI controls, to control the five separate satellite images. There was even a new menu heading on the client, to organize the new set of satellite image controls.

All these changes were accomplished in a matter of hours, without changing any compiled code. While this might not have been the optimally efficient solution for dealing with multiple satellite images, it was a solution that could have been implemented by an appropriately trained system administrator in less than a day.

5.2 Capabilities of an Evolvable Work-Centered System

Based on the observations described in Section 4, we present a list of "evolvability capabilities" of an evolvable work-centered system. Each of the items in this list represents one way such a system would be able to be changed, without resorting to bringing in programmers to implement the changes.

- *Bringing new data into the system.* Many of the change requests we saw with the WCSS-GWM were requests to provide new functionality for the user by making new data available to the user. The first step in satisfying such a request is simply to make the new data accessible to the system.

- *Adding new data to an existing display.* Once the data is accessible to the system, it needs to be made visible to the user in an appropriate way.

- *Receiving existing data from a new source.* One of the most common change requests we've seen resulting from factors outside the control of the users is a change in data source. Whether the existing data just isn't available any more or there has been a change in format, we need to be able to easily accommodate such changes.

- *Altering the way data is presented in an existing display.* Changing how data is presented in a display (e.g., color and symbology) is actually one of the easier types of system changes to accommodate. Many existing C2 systems already allow their users to "customize" their displays in this way.

- *Reviewing and altering the transformation and filtering rules of the Analysis Module.* Each of the decisions made by these rules must be understandable and transparent to the users of the operating organization. The behavior of the Analysis Module must be able to be changed by the user; there must also be support for the user to easily evaluate whether his change has had the desired effect.

- *Reviewing and altering the prioritization behavior of the Presentation Module.* Just as with the rules of the Analysis Module, the behavior of the Presentation Module must be understandable and easily modifiable.

- *Allowing integration with 'homegrown' tools/artifacts.* As users in the operating organization get ever more technically sophisticated, they begin to build for themselves spreadsheets and text files that systematize information that is not available in their standard systems. It would be ideal if our evolvable work-centered systems could end the need for such tools, by giving the users enough capability to change the work-centered system. Until our evolvable systems are flexible enough, though, it would be prudent to allow easy integration with such user-defined tools, by explicitly defining mechanisms for integration with spreadsheets and text document.

- *Supporting 'local override databases'.* One drawback to the use of existing C2 systems that we have observed relates to the currency of data. Standard systems are tied to standardized data sources. We have seen in multiple organizations situations in which users have knowledge of temporary data changes – unpublicized airfield closures, changes to preferred routing procedures, even late-breaking news of aircraft maintenance delays – which

just do not fit into their system. The systems do not provide easy mechanisms for temporary data changes; the effect is system displays must be disregarded. By explicitly allowing for a 'local override database' – a user-controlled database of critical knowledge they have that overrides standard data – our evolvable work-centered systems can make use of the detailed knowledge of the local expert.

5.3 Software Architecture and Technology for Building Evolvable Work-Centered Systems

As the development of the WCSS-GWM has progressed over the past few years we have taken some initial steps in exploring the capabilities required to evolve the WCSS-GWM to meet changing requirements. More recently we have been looking at extending these first steps to more formally support system evolution and more particularly at how the users of the system might be enabled to play a substantive role in this process. In pursuing this initiative, we have been working within the WCSS-GWM—the functions that support the evolution of the system are an intimate part of the WCSS-GWM.

This within-system approach raises an immediate question: Does system evolvability have to be built into the system up front or is there the possibility that the evolvability capabilities can be imported and adapted to an existing system? As we look at software technologies to support system evolution we are looking first at how to work from within a system design process, but at the same time thinking about how these capabilities might be made more generally available by developing capabilities that can be imported into an existing system.

When working from within the system, what differentiates the process of building evolvable work-centered systems is the meta-design task of identifying evolvability requirements, starting at the beginning of the design process and continuing throughout the design and implementation of the system. Based on our current WCSS-GWM experience, as described above, many areas that can be expected to require change have been identified. Starting from this experience base, as designers and users work together to define the requirements of the new system, requirements for capabilities to support system change can be developed in parallel. These evolvability potentials help to map out the space of possible changes, and so help provide criteria for evolvable system design decisions.

5.3.1 Architectural Support for Evolvable Systems

The architecture of an evolvable work-centered system must facilitate user-developed additions to system capabilities. This will involve an extensive set of user interface components through which a user will accomplish the desired changes. Data access, analysis, and presentation changes will require detailed representations of the relevant domain data objects. For presentation changes, detailed representations of the target display screen entities will be required as well. For many of the changes that developers

are confronted with today, new code must be provided to accomplish the change—that is, the new capabilities are procedure-driven rather than data-driven. While much of the thrust of this effort will be towards enabling data-driven change, there will be points at which new procedural steps will be required to accomplish desired changes.

5.3.2 Agents and Agent Templates

As new processing components are added to the system – new data acquisition, analysis or presentation tasks – the architecture must allow for the new tasks to be added in a manner that does not lead to harmful interactions between new and old tasks. At the same time, it must be easy to add new tasks that can communicate with and effectively take advantage of existing system capabilities. There must be facilities built into the architecture to support the test and evaluation of the new capabilities.

The D-OMAR agent-based system, on which WCSS-GWM has been built, provides an architecture that helps supports each of these requirements. Individual D-OMAR agents, working independently, provide encapsulated WCSS-GWM functionalities. A publish-subscribe protocol supports the communication among agents. The publish-subscribe protocol is used to coordinate the actions of agents working together on a common goal and to move data among agents. As new capabilities are added or existing capacities are refined, agent behaviors can be modified or new agents employed to meet the new requirements. The publish-subscribe protocol supports new agent access to existing agent capabilities.

In order to usefully add new functionality to the system, we need to provide building blocks that provide essential capabilities. Part of the job of designing an evolvable work-centered system is to develop a set of these building blocks. Based on our development experience with WCSS-GWM, we have identified two building block types.

In WCSS-GWM there are two recurring types of agents – data acquisition and analysis agents. Most of the data acquisition agents follow the same basic processing pattern – periodically try to get some data (from a website, from an FTP server, or from a database, for example), reformat the data that is received into a predefined XML format, and push the resulting file to the WCSS-GWM clients. Building on this process, the next step is to encapsulate this behavior in a standard "agent template." The goal will be to instantiate a new data acquisition agent "on the fly." There are many details about which the agent will need to know – location of the data server from which it will get data, protocol to use to get it (ftp, http, https, SQL), and filtering and formatting instructions to tell it what bits of data to write out to its output. Much of this can be provided using the data and information definitions as discussed in the next section.

WCSS-GWM analysis agents generally similarly follow one of two standard patterns. Analysis agents may be responsible for a particular air mission with operations such as watching for obstructions in its path. Conversely, they may be responsible for a particular hazard and watching for missions that will be affected. These particular agents have capabilities that are quite specific to the WCSS-GWM domain. As such, they are not

expected to be easily generalized to other evolvable work-centered systems. But the notion that an evolvable work-centered system will have patterns of agent behavior, which can be usefully abstracted into agent templates that can be instantiated on the fly, will provide a powerful mechanism for evolvability.

5.3.3 Defining the Data to Support Change

A key strategy in allowing new types of data to be easily brought into the system is the development of a structured description of the domain-level data and information objects in the system. Some of these information objects in the WCSS-GWM define air missions, airfields, and weather observations. For each type of object, we currently describe the key attributes of the objects that are used by the system, either for reasoning or for display. Some of the key attributes for air missions are mission identifier, origin airfield, destination airfield, and schedule; airfields contain attributes for identifier, latitude, longitude, and set of runways.

For each type of display provided by the system, we also maintain a structured description of the information objects visible in that display. This description identifies the attributes of the information object that are primary in the display – visible at all times in the display – as well as those attributes that are secondarily available in the display – perhaps only visible in a mouse-over, or by bringing up a pop-up display. So, for example, the origin and destination of an air mission is a primary attribute displayed by the WCSS-GWM map display, while the schedule for that mission is a secondary attribute, available in a mouse-over. When displaying a mission in a time-line display, the roles are reversed: schedule might become primary while origin and destination become secondary. Similar structured representations of information needs are maintained for each of the analysis tasks in the system.

The purpose of these structured descriptions of information and information needs is to serve as a basis for matching supply and demand for information as system requirements change. Yet, there will be times, during the life of a system, when the information model we describe here will be found incomplete or inaccurate—new data will become available or the format of existing data will change. Hence, there will need to be the capability to refine or add to the data or information object descriptions. Today this is the domain of the developer—it is an important capability that should be available at the user interface. Whether used there by a developer or a user will depend, in part, on the complexity of the particular data item.

Today these data definitions exist in the Java code for the WCSS-GWM. There are several languages and tools for developing information models of this type. Possible candidates for use in our future research effort include OWL (http://www.w3.org/TR/owl-features/) and Protege (http://protege.stanford.edu/).

5.3.4 Defining Process to Support Change

Just as data must be defined so that it can be operated on to accomplish change, there are points at which it is process that must be specified to accomplish desired change.

We have described a prototypical evolvable work-centered system to have two modules in which decisions are made by the software. The Analysis module uses rule-based and algorithmic techniques to transform raw data into higher-level information of interest to the user. The Presentation module is responsible for prioritizing information for display to the user.

The reasons for the decisions made by these modules need to be visible and understandable to the user. By presenting the workings of these modules in rule form, the user can understand why certain decisions were made. These rules should also be editable by the user, and in fact, the system should be implemented to allow the user to make "what if" changes to rules. Such an approach would enable the user to experiment with the rules to see what effect his proposed change will have on the system.

5.4 Issues in Deploying Evolvable Work-Centered Systems

Before we can seriously propose to build and deploy evolvable work-centered systems there are several questions we need to be able to answer.

- Who (that is, what roles within the operating organization) will be responsible for making changes to the system?
- How can we identify which changes can be made to the system by non-developers, and which system change requests need to be handled in a more traditional way by software developers?
- What kinds of systems are appropriate to implement as evolvable systems? Are there systems for which this would be too dangerous, in terms of system reliability and safety?
- What procedures will be used for verification and validation of changes?

5.4.1 User-Organization Roles Relating to Evolvable Work-Centered Systems

There are a number of well-defined roles in an organization operating a traditionally-designed command and control system that work together through the development, testing, deployment, and operation of the system. An abbreviated set of such roles is described here – we don't need to describe all the possible roles; we'd just like to provide enough context so we can describe new roles that must be filled by an organization operating an evolvable work-centered system.

Software designers/developers implement the software. Independent verification and validation is performed by *testers*. *Users* are trained in operating the system by *trainers*. The system is maintained and monitored by *system administrators*.

In an organization operating an evolvable work-centered system, we see much the same structure, with some additional roles. Software is still initially implemented by *software designers/developers*. *Trainers* still train a set of *users* in operating the system and there is still a maintenance and monitoring role played by *system administrators*. In addition to these traditional roles, we see several new user roles relating to the operation and evolution of the system.

Two comments will be made before describing these new roles. First, multiple roles may be filled by a single person. Secondly, just because we describe these roles as new in the context of evolvable work-centered systems does not mean that these roles are entirely new, and have never before been filled in organizations operating traditionally designed systems. In fact, our team has seen each of these roles filled, generally in an informal assignment, in our target airlift service organization. What is new here is calling these roles out to the operating organization as roles they must fill, in formal assignment, with their personnel.

In addition to the roles described above, we envision the following roles:

- The *Expert User*, who understands the work and the system well enough that he understands not only what the system does, but can envision ways in which the system could be changed to help users perform the work better. Not only does the expert user informally assist and train novice users, but the expert user is likely the primary source of suggested system change requests.

- The *Local Data Expert* is the primary human memory of all the data necessary to support the work that is not easily represented in the system, or should temporarily override data that is in the system. While not every system needs such an expert, we have observed this role in at least three instances in our target airlift service organization. This expert is the person who knows who to call (or is the person who is called) to get information about temporary unscheduled airfield closures, or changes to preferred air routings, for example.

- The *Local Tools Expert* is the person who is most likely to make changes to the system. We have observed this role in the last few years as our users come to the job with more and more computer expertise. They use automation in the form of spreadsheet or text-editor macros to provide functionality their current systems just cannot do. In the context of the evolvable work-centered system, hopefully this expert will be evolving the system to provide new functionality instead of using home-grown automation tools.

5.4.2 Identifying Candidates for Evolvable Work-Centered Systems

Not every system is a good candidate for design as an evolvable work-centered system. The fact that system changes can be made by end users, without a full formal testing

cycle, places some limitations on the usefulness of the category of evolvable work-centered systems. In fact, the very statement that changes can be made to a system in place, without formal testing must strike fear into the hearts of some. But we can try to characterize the class of systems which we can safely implement as evolvable systems.

Even current command and control systems often provide some level of user customization of their user interface. They typically allow users to redefine how certain data elements are to be displayed – changing colors, line types and widths. The reason it is allowable for users to make these changes at will is that it is understood that these changes generally do not compromise the effectiveness of the system. Having identified one case where it is permissible for users to make modifications, we can try to "push the envelope" and ask ourselves what other kinds of changes can safely be made by users.

We maintain that there is a class of command and control systems that can safely be made evolvable. This is a class of systems whose primary purpose is the faithful display and integration of information. The system may contain some amount of rule-based or algorithmic processing to provide automated aiding, but any decisions are being made by the user, not by the system. Such a system would not include display of results of significant computational processing performed by the system – changes made to such a computational process would be unable to be validated by a user without a formal test suite.

As experience is gained with implementation of evolvable work-centered systems, we foresee the ability to identify certain modules or algorithms of a system as "locked" – not to be touched by users for fear of violating the integrity of system results. By advancing our understanding of what components can safely be modified by users and what components must remain untouched, we will reach the point where we can bring some level of evolvability to nearly any command and control system.

5.4.3 Testing and Validation of Evolvable Work-Centered Systems

Providing appropriate support for testing and validation is a critical step in designing and implementing evolvable work-centered systems. The goal is to enable the operating organization to make its own modifications to the system. If they cannot properly test and validate their system modifications, they will refuse to make those modifications, and any benefits in the evolvability of the system will be lost.

In fact, preparing the system for independent testing and validation by users will become one of the core tasks performed by developers all through the development cycle. We take some ideas from the recent practice of Extreme Programming (Beck, 2004). That is, no software module should be written without also writing a test for it. Preferably, the test will be an automated test, and will include known data with which the test will be run.

For purposes of developing an evolvable work-centered system, we modify these rules a little. Every software component that is visible to a user (or system administrator) – every display, every agent, every automated processing algorithm – must have one or more test procedures attached to it and developed alongside it, including canonical sets of data upon which the test will operate. Even a data acquisition agent, which normally produces only formatted data files and/or database records, will be developed to have a test mode in which it will produce visible output.

When an end user or system administrator makes a system change, at least two kinds of test procedures will be run. The first is the unit test, using the test procedure for the component that has been changed and comparing the output on the standard test data set for that component with the expected results. A successful test here says that the component has been successfully modified, and the modified component is working satisfactorily.

The second type of test procedure is a regression test, in which the effect of the modified module on the entire system is validated. To perform this test the entire system is executed, with the modified module, side-by-side with the unmodified system so comparison can be made. Testing in this way will identify problems arising from faulty interactions and interfaces between the newly modified module and existing modules.

The need to operate these second types of tests implies a certain amount of system infrastructure devoted to modification and testing. In fact, the requirement that the unmodified system and the modified system be able to be run simultaneously on the same data stream, without interfering with each other, does put some new requirements on the structure and sizing of the work-centered system. Either the software needs to be structured in such a way that multiple processes can be run on the same server without interference, or we need to allow for a separate testing suite of server hardware. It should be noted that this requirement is not outlandish – several military command and control systems being developed in recent years have had similar requirements levied upon them.

6.0 Discussion

In this paper we advance the thesis that for a system to remain 'work-centered' over time it must not only support the elements of work identified at a fixed point in time but also include provisions to accommodate change. The goal is to develop systems that not merely allow a user to tailor or customize the interface to meet short-term local requirements but to provide facilities that enable the user community to evolve the entire software structure so as to be able adapt to the changing demands of the world – *evolvable work-centered support systems.* We believe that such an aim is achievable and have pointed to some promising software directions.

Our observations and proposals are similar to, but distinct from, some related concepts that have been put forth by others. For example, a number of researchers have noted that users will informally tailor the design of their systems and work practices to better meet

the local demands of the situation. This has been referred to as 'finishing the design' (Rasmussen & Goodstein, 1987; Vicente, 1999; Mumaw et al., 2000; Vicente et. al., 2001). Vicente (1999) has argued for the importance of creating systems that afford the potential for productive adaptation to enable users to 'finish the design' locally in response to the situated context of work. Our findings and conclusions are consistent with Vicente's proposal. They extend the ideas by emphasizing that the demands of the world are not fixed but will change over time. Thus, 'finishing the design' is not merely a matter of responding to specific local conditions but entails adapting systems so as to keep pace with a constantly evolving world – in that sense the design is never really 'finished'. A related implication is that grounding a system design in a work domain analysis as part of a cognitive work analysis (Vicente, 1999), while necessary, is not, in itself, sufficient to insure that a system will provide the flexibility for productive adaptation. We contend that systems need to explicitly incorporate mechanisms to enable users to adapt the system to evolving requirements.

Our findings and conclusions also share similarity with the concept of the *'Task-Artifact Cycle'* (Carroll and Rosson, 1992). A number of researchers have pointed out that when new technology is introduced it can have unanticipated reverberations on the field of practice (Carrol and Rosson, 1992; Woods and Dekker, 2000). The new system may afford new possibilities recognized by the user community that result in it being used in ways that had not been anticipated by the system designers. Clearly the emergence of new, unanticipated users and uses for the WCSS-GWM system that we experienced, partly exemplifies this 'Task-Artifact Cycle'. However, as we document in Sections 4, some of the need for adaptation we observed reflected changes in the work environment, and could not be accounted for by the 'Task-Artifact Cycle'.

Our proposal for moving toward evolvable work-centered support systems shares commonalities with recent calls in the computer-human interaction community to move toward *End-User Development* (EUD) systems (Fischer, Giaccardi, Ye, Sutcliffe & Mehandjiev ,2004; Fischer and Giaccardi, in press). The goal of EUD is to develop tools to enable end-users to adapt and further develop applications to meet evolving requirements. It has it roots in early calls to enable users to create customizations, extensions, and applications so as to address unanticipated requirements (Mackay, 1990; Nardi, 1993). Fischer and his colleagues (2004; in press) have argued for the importance of developing meta-design approaches that create open systems that can be modified by their users and evolve over time. EUDs range from systems that provide for modest user modifiability to systems that have end-user programming features (e.g., open source code).

A distinguishing feature of evolvable work-centered support systems is that it's very derivation is based on a work-centered perspective. We applied the same work-centered design methodology, grounded in an analysis in the demands of work, to derive the requirements for evolvable work-centered support systems. A conclusion of our work-centered analysis is that the ability to evolve in the hands of users steeped in the context of work is fundamental to work-centered support.

Acknowledgements
The WCSS-GWM system development and evolvable systems research documented in this paper was sponsored by the Air Force Research Laboratory Human Effectiveness Directorate.

References

Beck, Kent and Andres, Cynthia (2004). *Extreme Programming Explained 2nd Edition.* Reading, MA: Addison-Wesley.

Carroll, J. M. and Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario, *ACM Transactions on Information Systems*, 10, 181-212.

Deutsch, S. E. (1998). Interdisciplinary foundations for multiple-task human performance modeling in D-OMAR. *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 303-308). Mahwah, NJ: Lawrence Erlbaum Associates.

Dittrich, Y. and Lindeberg, O. (2003). Designing for Changing Work and Business Practices. In Patel, N. (Ed.) Adaptive Evolutionary Information Systems. IDEA Group Publishing.

Elm, W.C., Potter, S.S., Gualtieri, J.W., Roth, E.M., & Easter, J.R. (2003). Applied Cognitive Work Analysis: A pragmatic Methodology for Designing Revolutionary Cognitive Affordances. In E. Hollnagel (Ed) *Handbook for Cognitive Task Design*. London: Lawrence Erlbaum Associates, Inc.

Eggleston, R.G. (2003). Work-Centered Design: A Cognitive Engineering Approach to System Design. In *Proceedings of the Human Factors and Ergonomic Society 47th Annual Meeting, Denver, CO, Oct 13-18, 2003.* (pp. 263-267).

Eggleston, R. G., Roth, E. M. and Scott, R. A. (2003). A framework for work-centered product evaluation. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting.* (pp. 503 – 507). Santa Monica, CA: Human Factors and Ergonomics Society.

Eggleston, R.G., Young, M.J., and Whitaker, R.D. (2000). Work-Centered Support System Technology: A New Interface Client Technology for the Battlespace Infosphere. *Proceedings of NEACON 2000, Dayton OH, 10-12 October, 2000.*

Eggleston, R. G. and Whitaker, R. D. (2002). Work-centered Support System Design: Using Frames to Reduce Work Complexity. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting.* Santa Monica, CA: Human Factors and Ergonomics Society.

Fischer, G. and Giaccardi, E. (in press). Meta-Design: A framework for the future of end-user development. In Lieberman, H., Paterno, F., Wulf, V. (Eds) *End User Development – Empowering People to Flexibly Employ Advanced Information and Communication Technology*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G. & Mehandjiev, N. (2004). Meta-design: A manifesto for end-user development. *Communications of the ACM*, September 2004, vol. 47, 9, 33 -37.

Kirwan, B. and Ainsworth, L. K. (1992). *A Guide To Task Analysis*. Bristol, PA: Taylor and Francis Inc.

Mackay, W. E. (1990). Users and customizable software: A co-adaptive phenomenon, Sloan School for Management Dissertation, Massachusetts Institute of Technology.

Militello, L. G. and Hutton, R. J. B. (1998). Applied cognitive task analysis (ACTA): a practitioner's toolkit for understanding cognitive task demands. *Ergonomics*, 41, 1618-1641.

Mumaw, R. J., Roth, E. M., Vicente, K. J. & Burns, C. M. (2000). There is more to monitoring a nuclear power plant than meets the eye. *Human Factors*, vol 42, 1, 36-55.

Nardi, B. A. (1993). A small matter of programming, The MIT Press, Cambridge, MA.

Roth, E. M., Malin, J. T. , & Schreckenghost, D. L. (1997). Paradigms for Intelligent Interface Design. In M. Helander, T. Landauer & P. Prabhu (Eds.) Handbook of Human-Computer Interaction (2nd edition), Amsterdam: North-Holland. (pp. 1177-1201).

Roth, E. M. and Patterson, E. S. (2005). Using observational study as a tool for discovery: Uncovering cognitive and collaborative demands and adaptive strategies. In Montgomery, H., Lipshitz, R., & Brehmer, B. (Eds.) *How professionals make decisions*. (pp. 379 – 393) Mahwah, NJ: Lawrence Erlbaum Associates.

Sarter, N. B. and Woods, D. D. (1995). How in the world did we ever get into that mode? Mode errors and awareness in supervisory control. *Human Factors, 37*, 5-19.

Scott, R., Roth, E. M., Deutsch, S. E., Malchiodi, E., Kazmierczak, T., Eggleston, R., Kuper S. R., and Whitaker, R. (in press). Work-Centered Support Systems: A Human-Centered Approach to Intelligent System Design. IEEE Intelligent Systems.

Scott, R., Roth, E. M., Deutsch, S. E., Malchiodi, E., Kazmierczak, T., Eggleston, R. G., Kuper, S. R. and Whitaker, R. (2002). Using Software Agents in a Work Centered Support System for Weather Forecasting and Monitoring. In *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting. (pp.433-437)* Santa Monica, CA: Human Factors and Ergonomics Society.

Seminara, J. L., Gonzalez, W. R. & Parsons, S. O. (1977). Human factors review of nuclear power plant control room designs (Report No. EPRI NP-309). Palo Alto, CA: Electric Power Research Institute.

Vicente, K. J., Roth, E. M., Mumaw, R. J. (2001). How do operators monitor a complex, dynamic work domain? The impact of control room technology. *International Journal of Human Computer Studies*, 54, 831-856. Available online at: http://www.idealibrary.com

Wampler, J. L., Stilson, M., Thomas-Meyers, G., Roth, E., Scott, R. and Whitaker, R. (in press). Cognitive Work Aids for C2 Planning: Actionable Information to Support

Operational Decision Making. In *10th International Command and Control Research and Technology Symposium: The Future of C2.*
Woods, D. D. (1998). Designs are hypotheses about how artifacts shape cognition and collaboration, *Ergonomics*, 41, 168-173.
Woods, D. D. and Dekker, S. (2000). Anticipating the effects of technological change: a new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science*, 1, 272-282.

CHAPTER II.
Collaboration

This chapter will address our WIDE 6.2 work insofar as it relates to analyzing the prospects for facilitating collaboration among TACC staff members, TACC units and within TACC overall. These topics were also studied in the prior GAMAT Phase II project, and the results were reported in 2004. Owing to that precedent - and the significant overlap between its results and those obtained under WIDE 6.2 - this chapter will focus on topics and issues not already documented in that prior project.

The Importance of Collaboration in WCD Practice

The issue of inter-worker collaboration is intrinsically important in designing information technology (IT) applications for organizations (and sub-units thereof). The goal of work-centered design (WCD) is to design IT applications which are configured to reflect the operational conditions of a work activity as it is actually performed in daily practice. This requires WCD designers to generate their WCSS designs on a foundation of deep knowledge of the workplace and the work activities. One of the key components of the WCD designers' required knowledge base is "social knowledge' - i.e., knowledge of the workplace social environment, policies, relationships, collaboration requirements, etc.

No matter how precisely prescribed or rigorously structured a work process may be, its actual performance will be influenced and guided to some extent by factors that can only be ascribed to the workplace social milieu. Ambiguities or even mysteries concerning why work performance isn't as good as it could be are often resolved once you identify and analyze factors relating to how the workers interact. Sometimes the manner in which work is currently accomplished can only be explained in terms of workplace characteristics whose origin has more to do with the organization's constitution as a social environment than its configuration as a functional entity.

The importance of social factors in addressing work practices has long been acknowledge in WCSS projects and our ongoing formulation of WCD methodology. For example, WCD theory (cf. Eggleston, 2005) has long prescribed four key elements of work activity that must be taken into account:

- *Problem solving/decision making* - the selection of options and actions based on discerned states of the work subject matter versus desired states and outcomes.

- *Collaboration* - the negotiation, coordination, and conduct of one's own work activities in the context of others' relevant work activities.

- *Work product development* - the creation and / or refinement of a specific and tangible artifact whose generation is the objective of the task at hand.

- *Work management* - the monitoring, organization, and / or manipulation of the overall work process or its constituent activities so as to render them at least tractable and at most efficient and effective as a set.

Two of these four key aspects of work are tightly intertwined with workplace social processes and structures. Work management activities are typically motivated and guided by mandates, policies and conditions originating in social interactions among the work force. The whole notion of 'collaboration' is, of course, a purely 'social' issue.

In the portion of WCD that we term 'work domain analysis', a number of work ecology features are targeted for identification, documentation, and analysis. Some of these are clearly social in nature, such as:

- Organizational context (collaborative links; command & control)
- Parties / units from whom work is accepted
- Parties / units to whom work is passed along
- Parties / units with whom the user does or may confer
- Means employed for communicating the course of a particular work activity
- Means employed for coordinating one's task activities with peers
- Means employed for reporting one's task data to superiors
- Means employed for allocating and assigning tasks to subordinates

As we have accreted a working knowledge of TACC operations through the years, we have come to be able to visualize the generally loosely-coupled sorts of collaborative practices extant in TACC. In the last 2 years, our work knowledge capture and analyses focusing on Execution Cell operations have given us a basis for understanding the most tightly-coupled collaborative work setting within TACC.

In undertaking consideration of collaborative IT support for TACC, we must be careful. 'Collaboration' can be trivialized into a generic 'good thing'. In colloquial usage, the term carries the connotation of 'multiple people working together on a common work product'. In some sense, and at some level of granularity, all organizations can be construed as relying upon 'collaboration' in this sense. However, the prospects for fostering more or better collaboration within an organization are not uniform or universal. For one thing, some work processes are more amenable to collective or group cooperative tasking than others. This is the case within TACC - at least to the extent that certain functions are performed by specialists. Moreover, some organizations may have sound reasons (e.g., security; resource issues) which lead them to enforce relatively compartmentalized approaches to work processes of joint production by multiple players. As a military command and control center, TACC is one such organization.

Topical Background: CSCW and Groupware

The most widely-used label for research and development on collaborative applications of IT is *computer-supported cooperative work (CSCW)*. This title was coined by Irene Greif and Paul Cashman in 1984 as a marketing tag for a vision of integrated office IT support (cf. Greif, 1988). Proponents question the precise boundaries of this research and development area (e.g., Bannon & Schmidt, 1989), though none question the value of the issues addressed therein.

The title of CSCW represents "...a shorthand way of referring to a set of concerns about supporting multiple individuals working together with computer systems." (Bannon & Schmidt, 1989, p. 358). It can generally be said that CSCW pertains to the overall field of supporting task-oriented teams with information technology, while groupware refers to those products applied in providing such support. Johansen (1988) listed CSCW as the leading candidate among a set of 14 terms in use to describe the emerging field of research. The other 13 were:

- Technological support for work group collaboration
- Collaborative systems
- Workgroup computing
- Group decision support systems (GDSS)
- Interpersonal computing
- Departmental computing
- Augmented knowledge workshops
- CAC / CMC (Computer-assisted / -mediated communications)
- Group Process Support System
- Teamware
- Decision Conferences
- Coordination Technology

As a standalone research field or discipline, it would seem that CSCW peaked in the mid- to late-1990's. Since 1996 the number of products specifically characterized as being part of CSCW has been in steady decline. This is not to say that CSCW is 'dead'. The issues and topics to which CSCW was originally dedicated are if anything more important today than they were two decades ago. Analysis indicates that work previously characterized as pure CSCW has increasingly migrated to the category of applied IT studies termed *human-computer interaction (HCI)*. (cf. Horn *et al.*, 2004).

Characterizing the Activities being Studied

CSCW researchers invested much time and effort in the circumscription, categorization, and analysis of their focal subject matter during the period from the late 1980's into the early 1990's. The theoretical foundations laid out during that period remain viable bases for addressing collaborative technologies. The reason is that during the most recent

decade research and development efforts have been directed toward generating groupware capabilities and migrating these capabilities to keep up with the progressive evolution from mainframe-based systems to proprietary LAN-based systems to generic WAN-based systems and eventually the World Wide Web. During this most recent period no significant revisions to CSCW's original theory base have occurred.

Of the many taxonomies and classification schemes developed to describe cooperative work activities, one of the simpler ones will suffice for our present purposes. This is the trinary framework introduced by De Michelis (1990), who addressed cooperative work in such a manner as to provide a means for discussing specific software applications. This capability for mapping classes of work activity onto collaborative IT applications ('groupware') is not a feature of many other work activity schemata in CSCW. De Michelis' approach concentrated on the manner of cooperative activity, rather than on a comprehensive definition of cooperation itself. In fact, De Michelis did not attempt to define what he means by "cooperation". Instead, he noted a trend toward the use of task-directed groups in modern enterprises and claims those groups are "...defined by the pattern of commitments that group members make with each other and with third parties." (p. 2) Having established this focus, De Michelis proceeds to delineate three different categories of cooperation:

- *Coordination* is that process by which group members organize and/or synchronize their actions within the framework of a task, regardless of whether or not they literally work together in accomplishing that task.

- *Collaboration* consists of those activities through which multiple actors work together on a given task. In other words, collaboration denotes that form of work activity in which multiple actors must interact and jointly generate a work product.

- *Co-decision* is an extended form of collaboration in which the task is reaching a decision. Co-decision therefore connotes cooperative efforts toward the end of decision making. Loosely speaking, co-decision can be construed as 'cognitive collaboration'.

Characterizing the Application Context

The most common label for IT applications designed specifically to support collaborative activities is *groupware.* The label was coined in 1978 by Johnson-Lenz & Johnson-Lenz (cf. 1992, p. 130) to denote a combination of two factors:

- intentional *group* processes and procedures to achieve specific purposes
- soft*ware* tools designed to support and facilitate the group's work

There are a variety of paths leading to historical interest in groupware, and one might select any of a number of places to begin tracing its history. For the purpose of brevity,

the most important stream of work is that of Doug Engelbart, who is credited with many of the innovations which now make computers easier to use. These innovations are notable enough. However, it is Engelbart's (1988a; 1988b) overall vision of how computers can be employed in organizations which both sets the context for these individual achievements and establishes him as a key source of inspiration and guidance for subsequent research into collaborative computing.

Engelbart's vision was one in which workers deal with information rather than with physical goods. Such workers (*knowledge workers*) not only manipulate and manufacture data; they create knowledge of the task, the means for achieving that task, and of the work milieu in which they operate. Shared information environments provide the settings within which knowledge workers can augment as well as mutually pool knowledge, and individual workstations would allow easy access to users. Some key features in Engelbart's vision are:

- access to computers for all workers (including easy usability);
- linkages among all workers within an organization via telecommunications;
- storage of the organization's "knowledge" within this shared electronic environment; and
- the means by which the ongoing "knowledge" relating to operations can accrete to the shared environment.

The key concept here is the availability of a 'shared information space' within an organization. Bannon and Schmidt (1989) identify sharing within mutually-accessible information space to be a definitive characteristic of CSCW. Similarly, De Michelis (1990) cites "information sharing" as the key support need in collaborative activity. Bannon (1991) states the point even more strongly by calling shared information spaces the single most important component of a collaborative IT capability.

A good starting place for categorizing the types of work support tools developed under the aegis of CSCW or 'groupware' is a taxonomy developed by Johansen (e.g., 1988). This subdivides the categories of collaborative work and associated groupware applications in terms of time and space parameters. Johansen's taxonomy is illustrated in Table 3.

Table 3: A Taxonomy of Collaborative Work Activities
(Adapted from Johansen, 1988)

	SAME TIME	DIFFERENT TIMES
SAME PLACE	• Face-to-face meetings • Working sessions • Conferences and workshops • Intra-shift co-located work activities	• Inter-shift work processes • Administrative oversight • Feed-forward production processes
DIFFERENT PLACES	• Teleconferences • Video conferences • Conference calls	• Email • Bulletin boards • Forms management • Voice mail • Structured messaging

Practical Background (Our WCSS Work)

Beginning in 1999, AFRL has conducted a series of projects focused on TACC mission processes and tasks (HISA, IFM, GAMAT Phase I, GAMAT Phase II). Each of the pre-FY04 projects were directed toward one or another particular function or position within the many comprising TACC (mission planners in HISA; FM's in IFM; WX staff in GAMAT Phase I). With GAMAT Phase II we more or less 'filled in the final gaps' in our perusal of the core TACC mission process path by examining the flight planners, the DIP planners, and the recently-implemented Execution Cell. This is not to say that we have a complete understanding of the complexities of TACC and its operations. However, it is fair to say that with the close of the GAMAT Phase II project we had accrued a top-level knowledge based spanning all the primary roles, positions, and activities comprising the mission operations process path.

This synoptic view of TACC operations was achieved during the past 1 - 2 years. This WIDE 6.2 project was conducted in the wake of the FY03 GAMAT Phase II effort - the fourth consecutive project in which AFRL researchers had studied TACC operations and prescribed WCSS design concepts. It was in the latter stages of GAMAT Phase II that we finally obtained a working overview of TACC mission operations and began re-evaluating our prior work and future prospects with respect to supporting broader collaborative processes in the target organization. This progression can be illustrated with regard to the foci and the products of our AFRL WCSS projects up through this reporting period, as summarized in Table 4 below.

Table 4: Overview of Subject Matter and Design Concepts: FY99 - FY05

PROJECT	SUBJECTS STUDIED	DESIGN CONCEPTS
HISA FY99-FY00	• Channel mission planning • MOG • Resource competition among planners	• Port Viewer[1] • Conflict Summary • 'Smart Lieutenant' • Structured Listing of Pending Missions to afford SA over the pending workstream[2]
IFM FY01	• New integrated flight management (IFM) mode of work • Flight Managers (FM's) • Utility of IMT Dashboard	• Mission Summary Display (concise 'to-do list' with alert cues) • Flight Planning Palette[3]
GAMAT Phase I FY01-FY02	• WX forecasting and WX support to TACC • WX 'back' and 'front' shops	• GWM-WCSS[4] • Sortie Palette[5]
GAMAT Phase II FY02-FY03	• Flight planning • Execution Cell processes • DIP planning	• GWM-WCSS (refined and extended) • Flight Visualization Tool (FVT) • DIP Summary Palette
WIDE FY04 - FY05	• Execution Cell • Mission planning • DO's / Seniors	• Individual Timeline Display • Multi-Mission Timeline Display

As illustrated in the table, up through GAMAT Phase I our AFRL projects had focused on a particular position or function within the overall TACC operational setting. As a result, each of the design concepts generated within these projects ended up being tailored to the tightly-circumscribed context each project's objectives and knowledge acquisition required. This compartmentalized focus did not, however, persist. Each of those prior projects' WCSS design concepts has been carried forward toward deployment and everyday usage. In some cases these ongoing developments have remained framed within the original scope of their development and presentation. Some, however, have evolved toward the more general scope of application which had been envisioned in their formulation. Examples of such wider operational application include the following:

[1] Multiple prototypes and applications based on the original Port Viewer concept have been produced since 1999. Additional labels for these descendants of that inaugural HISA product include: 'MOG Tool'; 'MOG Viewer'; and 'HISA Tool'.

[2] This concept would later be refined and recommended anew in the form of the Mission Summary Display in FY01's IFM project.

[3] The Flight Planning Palette was introduced as a paper concept in the IFM project's final briefing (March 2001). As we learned in FY03, the FM staff had accepted this recommended concept and taken action to have it developed using local resources. The result - termed the 'Sortie Manager' - is now an application available for use by the FM's.

[4] The GWM-WCSS was also known (at various times and by various people) as *Weather Management Tool (WMT)*, the 'Weather Tool', etc. The original prototype (developed and refined by Dr. Ron Scott and his BBN colleagues) has become a deployed application within TACC. As such, the GWM-WCSS is the sole example to date of an AFRL team's prototype which has itself migrated into deployment (as opposed to being re-implemented for deployment).

[5] The Sortie Palette developed in the GAMAT project was an adaptation of the IFM project's Mission Summary Display concept implemented as an adjunct to the GWM-WCSS.

- HISA's Port Viewer was originally one of three 'viewers' comprising a comprehensive mission visualization tool. It was extracted from this larger set and tailored to address the MOG problem which our TACC customers specified as our research focus. In its various (but relatively unvarying) incarnations, this visualization tool has evolved into a general purpose 'gadget' employed by more than one role within TACC.

- The IFM project's Mission Summary Display was conceived and recommended as both (a) a specific remedy to on-screen visual overload problems identified with the IMT Dashboard and (b) a general top-level situation awareness aid of general utility to all involved in TACC mission processing operations. After the close of the IFM project, this concept was carried forward as (a) a specific display option to be implemented in the oncoming GDSS-2 application and (b) a general mission SA aid associated with the specific tool developed for the WX staff (the 'Sortie Palette' embedded in the GWM-WCSS).

- The IFM project's Flight Planning Palette (FPP) was not carried forward for prototyping in subsequent AFRL projects. However, the intended users (the FM's) took it upon themselves to pursue the concept, and they succeeded in getting it developed locally in the form of what's now called the 'Sortie Manager'. Once we reached the point where we could envision a general mission planning support suite aiding multiple roles within TACC, the checklist and procedural SA capabilities of the FPP recommended themselves anew as candidate features for WCSS designed for broader collaborative usage.

- The GWM-WCSS prototype has evolved into a general purpose WX visualization aid which has been deployed for use by TACC staffers outside the population of WX staffers for whom it was originally designed. Flight managers and others have found it useful for visualizing and analyzing weather information in support of their planning and monitoring tasks.

Although our prior design recommendations and concepts were contextualized with respect to narrowly-delineated tasks or requirements, they were conceived to be more generally useful - at least in theory. So long as we continued to focus on a particular position or function, our design and prototype products could be generated with sole regard to the particulars of the context at hand and not the generalities of TACC as a whole. By the time of GAMAT Phase II, we found ourselves having to confront the relationship between the specifics of previous design concepts and the practical reality of having to provide support to multiple (or all) TACC roles and functions comprising the mission operations process path.

The GAMAT Phase II effort marked the first of our four projects to date in which the AFRL team was asked to consider the overall suite of IT support tools available to TACC in light of the overall TACC organization, work process, and workflow. This more general scope of consideration required some adaptations in the number and the details of

our suite of WCSS design concepts. These adaptations became necessary because the 'granularity' at which our earlier design concepts were framed was narrower than the 'granularity' at which we now had to assess the concepts' viability with respect to this more general scope.

As a result of this assessment, it was determined that the basic set of capabilities provided in our earlier prototypes and concepts was still viable for overall TACC operations. However, some of the capabilities were more 'generic' than may be evident in their current associations with one or another specific position, deployed functionality, and / or prototype. The implication was that some features or capabilities needed to be re-framed and reallocated into more general forms as we expanded our scope of consideration from individual units to all of TACC. A summary of the recommended adjustments as of FY03 is given in Table 5.

Table 5: Adapting Prior WCSS Capabilities to Address TACC-Wide Operations and Team Collaboration Requirements
(Adapted from the GAMAT Phase II Draft Final Report)

CAPABILITY	CURRENT STATUS	ADAPTATION REQUIRED
Port / MOG Visualization	• Currently provided by multiple MOG / Port Viewer prototypes and applications	• Generalize the currently MOG-specific focus to provide visualization for a variety of Port / Airfield factors.
Summary Workstream Situation Awareness	• Currently demonstrated in the Sortie Palette element associated with the GWM-WCSS • TACC staff not using GWM-WCSS must monitor the pending workstream via (e.g.) IMT or GDSS.	• Decouple the Sortie Palette from its current placement within the GWM-WCSS • Establish the Sortie Palette (or equivalent) as a discrete application / aid within an overall TACC collaborative support suite • Elevate this workstream aid to universal access across the TACC team
Geospatial Visualization	• Currently provided in the GWM-WCSS prototype • Offers composite visualization for weather and route / mission elements • Current prototype is tailored to WX needs and incorporates some features / priorities peculiar to WX shop	• Migrate the general or basic visualization capabilities to a discrete application provided a wider population of users • Specify the layers and options (analogous to those provided WX staff already) which need to be prioritized for flight-oriented visualization
Single Flight Planning Palette / Portal	• Currently available to FM's in the form of the Sortie Manager • Some features and options are peculiar to FM needs and functions	• Generate a more generic version of the Sortie Manager application • Decouple the FM-specific options and features for this more generic prototype
DIP Status for a Given Mission	• Currently available if one can find it in the free-form text stored in the given mission's Logbook (DAP) records • Current BBN ACT Tool provides prototype DIP processing tool, but doesn't provide general SA on DIP's to other TACC team members	• Generate a design concept for a generic DIP status display providing SA and master caution panel functions with respect to diplomatic clearances

The adaptations cited in Table 5 represent a 'repackaging' of the WCSS products' functionalities to permit a wider scope of deployment in support of the TACC operations team. The functionalities developed up through GAMAT Phase I can be seen as a somewhat piecemeal approach to identifying and mitigating TACC needs, driven by the piecemeal set of foci set for our AFRL team over the years. In moving forward to consider the entirety of the TACC operational team and their joint needs, we must re-evaluate and decide which functionalities are to be associated with which positions and roles. This translates in most cases to a process of generalizing functionalities created with respect to only one such position / role so as to reflect the commonalities of objectives and requirements within a broader TACC workforce.

The first specific outcome of this reorientation was the generalization of the GWM-WCSS into a Flight Visualization Tool concept during GAMAT Phase II. This broadened the application scope of the GWM-WCSS' geo-spatial capabilities to aid positions outside the TACC weather shop, such as the flight planners. The second product illustrative of this generalization is the timeline tool WCSS concept described in more detail in Chapter III. Both these WCSS support tools were conceptualized as general purpose aids whose utility could span both (a) the range of all TACC personnel participating in the transport mission process path and (b) the set of all TACC organizational units within which these personnel were positioned.

TACC's Organizational Structure and the Prospects for Collaboration

TACC is a large organization subdivided into several functional subunits, each of which is delineated with respect to particular roles within or aspects of the transport mission process path leading from initial mission planning through to execution. This organizational architecture compartmentalizes participating personnel within specialized subcomponents of what can be construed as a composite TACC team. This creates an obvious compartmentalization of functions and responsibilities along the transport mission process path.

In other words, TACC is currently configured in a way which does not officially mandate or even foster a style of work activity in which multiple staffers cooperate on a common work product in realtime or in near-realtime. TACC (as currently configured) functions on the basis of common effort, but not through any widespread 'collaboration' in the sense of closely coupled interactivity among different roles and positions. A cursory overview of TACC's 'collaboration profile' relative to Johansen's time and space taxonomy is given in Table 6.

Table 6: TACC Collaborative Work Activities
(Based on Johansen's taxonomy)

	SAME TIME	DIFFERENT TIMES
SAME PLACE	• Face-to-face meetings • Working sessions • Intra-unit / - office consultations and cooperation • Intra-shift handoffs / briefings • *Some* activities in the Execution Cell	• Work processing within a unit but performed across shifts
DIFFERENT PLACES	• APCC consultations with external logistics staff • Interactions with ATC • Interactions with wings / squadrons • Incoming calls from aircrews	• All activities in the TACC mission process path except for *some* activities in the Execution Cell

Certainly, there are particular junctures in the TACC process path where individuals within functional subunits (e.g., the flight planning shop; the former 'swimming pool' for the FM's) confer or cooperate. However, even at these finer-grained levels of granularity there are few persistent examples of multi-person 'collaboration' to be found. The primary exception to this claim is to be found in the Execution Cell (i.e., on the 'Ops Floor'), where a confederation of specialists work jointly to oversee missions in progress. Even in this case, however, the idea of a persistent sub-team working a particular mission is not the rule.

The fact remains that there are some activities within which subsidiary tasks are sufficiently well-defined as to be most efficiently accomplished by specialized individuals (or sets of individuals) operating in relative isolation from peer or parallel persons contributing to the same overall process or product. Even in such cases as these, there may still be grounds for promoting 'collaboration', but in a sense other than 'multiple people working on a common work product'.

This situation can occur when otherwise-compartmentalized functions, no matter how efficiently conducted in and of themselves, must contend with conditions and constraints relating to other peer functions (and / or external conditions mediated by peer functions). In such cases, the point is to ensure that each of the compartmentalized subunits do not separately and efficiently contribute to a work product which is 'wrong' (i.e., ineffective) when evaluated as a composite output. The usual label for keeping individually-*functioning* peer elements jointly-*informed* on what and how to accomplish the team's work is 'coordination'. This is generally the same as the work activity of the same title De Michelis (1990) incorporated in his cooperative work taxonomy.

By the same token, all these relatively compartmentalized individuals and subunits must make decisions whose correctness is predicated on what others are doing, have done, or will do. This can be seen as a TACC-wide exercise in what De Michelis (1990) labels 'co-decision'. Furthermore, each of these individuals must be prepared to modify mission plans in response to changes resulting from the actions of their peers (and, of course, external parties for whom peers are the primary points of contact). The single most

common resolvable breakdown condition cited over the years we've been studying TACC is that which occurs when one or another aspect of a mission's plan is invalidated without the person(s) responsible for necessary corrections knowing it. In other words, the solution path for optimizing TACC team operations is more along the lines of facilitating 'coordination' among individually-specialized elements than instituting literal 'collaboration' among them.

As a result, our AFRL WCSS projects have not focused on design concepts specifically tailored to a usage scenario within which multiple people 'collaborate' in realtime. Instead, we have concentrated on design concepts which permit multiple players (perhaps in widely-separated locations) to jointly view and / or manipulate relevant mission information. In other words, we have sought to support overall team work processing without configuring our designs such that team members must link up synchronously to exploit these design concepts. Conversely, our WCSS design concepts have historically been developed so as to avoid preventing or subverting realtime 'collaboration' among TACC team members. In other words, although we have deliberately avoided *forcing* realtime collaboration among TACC team members, we have deliberately left open the prospect of their doing so as they see fit.

The Central Collaborative Challenge: Dynamic Changes External to TACC / USAF

If there is a single compelling justification for introducing additional collaborative IT support in TACC, it would have to be based on what happens when things change. A recurring complaint we've encountered dating back to the HISA project in FY99 concerned the manner in which shifting circumstances outside the scope of direct inspection (and hence outside the immediate ken) of TACC staff can 'clobber' a mission. Such negative impacts can occur at any point along the planning - to - execution process path. Such changes can originate with any of the players associated with executing a mission, and they may pop up at any time. The range of such potential changes can be illustrated with the following examples cited during our FY03 KA activities on GAMAT Phase II:

- Change requests incoming from the pilot / aircrew
- Changes to cargo parameters (amount; weight, etc.)
- Changes relating to presence of Hazmat (e.g., late recognition that Hazmat is involved; Hazmat being substituted on a priority basis for some originally-intended cargo)
- Changes in DIP clearance viability or the regulatory bases for DIP clearances
- Changes in airfield availability or operating parameters (e.g., via NOTAM's)

- Changes in scheduling pertaining to assets assigned to the planned mission (e.g., aircrew, aircraft, specific cargo items)

- Changes in routing or scheduling relating to significant weather conditions (e.g., tropical storms)

- Changes in scheduling deriving from crew rest requirements

- Changes in aircraft availability deriving from maintenance requirements

In general, such dynamic changes can jeopardize the viability of a flight plan at any time during the planning / execution process. Such changes become particularly problematical as the time for mission launch draws near, because:

- It is only at this relatively late stage in the process that certain key facts and factors can be determined with any certainty.

- It is at this late stage that opportunities for timely adaptations capable of 'saving' a mission plan diminish or disappear.

- The delay until one can access newly-issued or updated information (e.g., new NOTAM's) typically equals or exceeds the planning horizon at this late stage.

- To the extent it is available at all, definitive information about remote (i.e., on-site) circumstances can sometimes only be obtained through direct contact with a relevant remote authority.

- Particularly during the period during which we've conducted our most recent projects (with multiple overseas theaters having to be supported) the opportunities for corrective replanning had become even more constrained owing to demands on USAF assets (e.g., aircraft, crews).

It is difficult to get a sense of what proportion of execution-ready flight plans typically have to be re-planned as mission launch time approaches. Anecdotal data is all we have been able to collect during our work knowledge capture efforts. Such data, though, indicates the proportion of pending mission products which must be modified during a given shift can run as high as approximately 25%. To deal with such changes under time pressure, TACC staffers must obtain relevant or current data from authoritative sources. For factors such as those addressed in this subsection, such authoritative sources are external to TACC itself.

This means that external contacts are an important tactic in trying to keep up with events and conditions. We have seen such external contacts employed in most of the TACC units we've studied over the last 4 years. In the IFM project (FY01) we observed FM's repeatedly taking the time to call ATC and on-site airfield centers to double-check on conditions, constraints, etc. In the GAMAT Phase I project (FY02) the staffers

performing the WX 'chartmaker' role consistently mentioned the occasional need to seek detailed on-site (e.g., airfield) data, even at the cost of a direct contact. For the flight planners we studied in GAMAT Phase II (FY03 - 04), email to and from Euro Control and / or UK ATC is a common channel used to double-check on conditions and changes for the all-important European airspace. The importance of such external contacts to the DIP planners is evidenced by their habit of maintaining 'contacts folders' containing key contact data.

Collaborative Information Technology (IT) Issues

The relative compartmentalization of TACC functions and personnel is reflected in a measure of compartmentalization in the IT support tools provided to and / or used by a given role or unit. As of FY03, only the flight managers (and / or anyone else) using the IMT Dashboard could claim to have persistent access to both a summary display of mission status parameters and automated support for alerts and warnings on pop-up conditions inimical to mission viability. Without this capability, TACC staffers may have to proactively access one or more IT assets to ascertain current mission parameters 'as planned' and / or additional data necessary to evaluating continued plan viability.

The result is a situation in which TACC staffers are expected to maintain situation awareness over a multi-dimensional problem space within which relevant variables / factors are mutually constraining. The absence of a 'one-stop' visualization or other display affording them comprehensive inspection of these mutually-constraining factors individually, much less as a composite, is one of the key deficiencies in the current TACC IT infrastructure.

The reason this constitutes a deficiency is that the current situation not only facilitates, but practically mandates, certain 'blind spots' along the TACC mission operations process path. By 'blind spot', we mean situations in which TACC staffer A lacks visibility (or ready access to visibility) on parameters or decision factors such as:

- Proactive modifications to an entire plan made by peer staffer(s) B (C, D,...) within TACC itself;

- Proactive modifications to one or another subsidiary plan element (e.g., DIP's; entry / exit times) enacted by peer staffer(s) B (C, D,...) within TACC, and having 'cascade' or derivative negative effects on staffer A's current object of work;

- Proactively-induced constraints or constraint-inducing conditions (e.g., denials of proposed plans on review; NOTAM's) deriving from actions by external parties (e.g., ATC); and / or

- Changes in status or prospects in the operational environment (e.g., weather, MOG) of which staffer A has no knowledge.

Summary: The Prospects for Collaborative IT Intervention in TACC

The circumstances outlined in the previous section would seem to argue for an effort to insert better collaborative capabilities within TACC. To be fair, however, it is difficult to assess the degree to which addition of dedicated collaborative IT capabilities are recommended, much less mandated. The degree of TACC worker access to the full range of TACC legacy systems is progressively growing. The arrival of GDSSII is going to change the types and depth of data products available to planners and other staffers throughout the mission process path.

TACC currently operates with a high degree of relative efficiency and effectiveness, given the state of sophistication of its IT infrastructure, the organizational constraints imposed by its military nature, and the level of its pending workload. It is difficult to claim that TACC is in dire need of additional IT infrastructure for the sole purpose of promoting collaboration interactions or general coordination (as those terms were defined earlier in this chapter).

As discussed earlier, the most critical source of negative situations mandating collaborative action in a short timeframe are those which are caused by parties or situations external to TACC. Meeting this most critical requirement is not something that can be done by renovating or innovating TACC's internal IT infrastructure. To make externally-directed connectivity and situation awareness more efficient and effective will require that IT resources shared among TACC and diverse external entities will have to be established, improved, or extended. To date, this scope of intervention has remained larger than the scope of study we have been allowed to pursue.

As such, this chapter must conclude with the same general conclusion as was rendered in the GAMAT Phase II report: There is no practicable basis for undertaking specific collaborative IT interventions until and unless our WCSS is capable of expanding the scope of its research and development beyond the boundaries of TACC per se. In the mean time, we expect the eventual development and (hopefully) deployment of our latest WCSS concepts - the Flight Visualization Tool and the timeline tool - will afford our TACC clients IT support sufficiently improved so as to partially mitigate the present stresses associated with time-critical collaborative activities.

CHAPTER III.
Work-Centered Analysis and Design Support

Introduction

One component of the WIDE 6.2 project work was to support design and development efforts being performed under the aegis of a 6.3 WIDE project. That development work is described in detail in the WIDE 6.3 project's final report documentation. This section of the WIDE 6.2 report will summarize the course and the products of the design support activities themselves. As such, this document will concentrate on the activities in which Randy Whitaker (NGIT) and Gina Thomas-Meyers (AFRL/HECS), operating under the aegis of WIDE 6.2, participated in a WIDE design team along with Emilie Roth (Roth Cognitive Engineering) and Ron Scott (BBN). Drs. Roth and Scott were working under the aegis of the WIDE 6.3 project.

During the period of the WIDE 6.2 project, this design team's efforts were focused on the formulation, design, and conceptual demonstration of a 'timeline tool'. The timeline tool is a work-centered visualization which plots transport mission features with respect to time. In other words, a timeline tool is intended to provide a user with the ability to display, analyze, and even manipulate mission parameters in terms of their being events and / or key points in time.

During the course of the WIDE 6.2 project, such a timeline tool WCSS was proposed, conceptualized, checked with respect to TACC customer needs, designed to a level of detail sufficient to support prototype development, and illustratively demonstrated to a TACC audience. At the time of the WIDE 6.2 project's conclusion, the timeline tool design specifications were sufficiently robust to serve as the basis for ongoing development and evaluation work. This ongoing work will be conducted under the aegis of the WIDE 6.3 project framework, which continues through FY05.

In the following sections, we shall present a summary history of the timeline tool design work and an overview of its products (i.e., the timeline tool design specifications). This presentation is intended to provide:

- A record of the WIDE 6.2 design support work as performed
- An illustrative case history of an actual WCD effort
- An overview of the timeline tool's design rationale
- A description of the design features incorporated into the timeline tool prototype concept

- Background explanations for the manner in which both our problem analysis and design deliberations led from concepts to concrete specifications

The thematic scope of this chapter will be delimited with respect to those activities to which the WIDE 6.2 tasks were specifically directed. These activities were circumscribed as 'design support'. In practice, this meant effort applied to project planning, knowledge acquisition, problem analysis, prototype conceptual design, and presentation of the design concepts to the TACC customers. Details of activities and products subsumed under the sibling WIDE 6.3 project will be provided in that project's final report documentation.

Basis for Focusing on Timeline Visualization in 2004

The rationale for undertaking the design and development of a timeline tool WCSS included criteria, observations, and reasoning which our WCSS team had inherited from earlier AMC projects, as well as similar bases arising within the context of the WIDE project itself. In the following subsections we shall review some of the themes which motivated our focus on a timeline WCSS during this reporting period.

The Criticality of 'Time' in TACC Operations

As far back as the HISA project (1999 - 2000) it had been noted that many of the problematical conditions and alerts based upon such conditions were correlated with time. A mission in progress is a sequence of actions and events that must happen in a certain order and meeting certain conditions to be successful. Flight arrivals and departures, flight progress through multiple controlled airspaces, and coordination among multiple flights are examples of factors which are most effectively referenced with respect to a temporal coordinate space.

Beyond this, TACC operations themselves are conducted under time pressures. The most important such pressures arise during the hours leading up to mission launch and continue throughout the period during which the mission is being executed. In other words, the temporal aspect of TACC mission planning and execution operations extends beyond the subject matter itself (i.e., a given mission) to the activities through which that subject matter is addressed, managed, and manipulated.

Addressing Two 'Dimensions' in TACC Operations

By the time that the GAMAT Phase II project had ended, our team had come to view TACC operations in terms of two 'dimensions'. The first is a 'horizontal' dimension, 'cutting across' TACC units and positions participating in the mission planning / execution process path. This 'horizontal' dimension is the one that needs to be considered in dealing with end-to-end process path support and the issues relating to how TACC's various roles and units can optimally collaborate.

The second figurative dimension of concern - the 'vertical' dimension - has to do with the chain of command and the organizational hierarchy within TACC. As we had learned in our earlier WCSS projects, TACC supervisory staff (e.g., seniors and duty officers) were at a disadvantage in trying to maintain summary situation awareness over multiple missions while relying exclusively on text-intensive data displays. One of the reasons we prioritized route visualization in GAMAT Phase II was a belief that coherent graphical route displays would be of as much (if not more) utility to supervisory staff as to planners and controllers themselves.

As we moved forward through the WIDE project, we had elected to make support for both these dimensions a key criterion for choosing our WCSS design and development objectives. We wanted our next WCSS product to be generic enough to be used by multiple roles / positions distributed 'horizontally' along the TACC process path as well as by multiple roles distributed 'vertically' through the immediate chain of operational command.

Generic Tools for a TACC-Wide WCSS Suite

Obtaining the desired payoffs along both the 'horizontal' and 'vertical' organizational dimensions required us to think in terms of generic WCSS tools that would be of utility to many roles. In the course of the GAMAT Phase I and Phase II projects (FY01 - FY04), our team had generated a geo-spatial visualization tool originally intended for the use of weather forecasters (the *GWM-WCSS: Global Weather Management Work-Centered Support System*). In Phase II of the GAMAT project, we had been examining the needs of flight planners - the staffers whose responsibility it is to construct feasible routing and flight plans to realize individual mission plans and mission requirements. We'd developed a generalized concept based on the GWM-WCSS which was labeled the *Flight Visualization Tool (FVT).*

The FVT was envisioned to serve as a central component of an expanded and redesigned WCSS toolkit. It was intended to provide geo-spatial visualization aid allowing TACC team members to efficiently and effectively review route information in a manner closely reflective of the actual flight being (or to be) executed. Like the GWM-WCSS, the FVT was designed around a central graphic map display augmented with a comprehensive set of overlay layers which can be mixed and matched to suit that user's particular needs.

The relevance of the FVT to our WIDE effort was twofold. First, we wanted our WIDE WCSS product to be configured to co-exist and interoperate with the FVT concept in a generalized TACC WCSS support suite. Second, the generality of the FVT served as a metaphor or exemplar for the breadth of utility we would seek to embed within a tool that addressed the temporal context rather than the geo-spatial.

Nomination of a Timeline Tool as the WIDE Developmental Focus: May 2004

The WIDE team held a technical interchange meeting (TIM) in Fairborn Ohio during mid-May 2004. At this meeting, Ron Scott gave a summary presentation of the rationale and the proposed format for a timeline tool display. This proposed display was to provide temporally-correlated visualization of missions at any stage throughout the planning-through-execution process path. Its display(s) would include representations for the events and constraints that affect the displayed missions. This proposed timeline WCSS would fuse data drawn from the major TACC information systems - e.g., CAMPS and GDSS/GDSSII - to afford users a unified picture of the mission itself (as contrasted with a set of data-centric information dumps).

Basic Features of the Proposed Timeline Tool

The initial proposal for the timeline tool included multiple features which would be carried forward into the conceptual and design phases of our WIDE project. The following subsections will summarize the fundamental design themes upon which the subsequent designs were based.

Graphical Representation of Temporally-Correlated Data

The proposed timeline tool was illustrated in our May 2004 TIM in terms of a display utilizing horizontal graphical units to portray events and other parameters plotted against a horizontal time index. Points or similar indicators would denote specific times on the time index. Elements exhibiting a duration would be denoted with a continuous horizontal line (or equivalent geometric coding) registered to indicate the period of applicability. As such, the proposed visualization approach closely mirrored the protocol used for representing airfield traffic flow and MOG (maximum-on-ground) conditions in the HISA Project's Port Viewer. This general presentational format is illustrated in Figure 3.

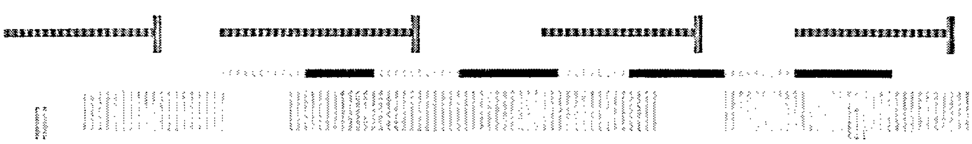


Figure 3: Initial Timeline Visualization Concept

Different types or forms of mission data would be allocated different lines. Relationships among these data types would be visible in terms of how the horizontal display elements aligned relative to each other. The idea was that such visual relationships could be coded to reasonably designate logical relationships. For example, if a period of foreign nation overflight required a diplomatic clearance (DIP), the horizontal graphical elements indicating the projected overflight period and the period of DIP viability would have to correlate in a certain way to connote a feasible mission plan. If the horizontal extent of the DIP viability 'bar' equaled or exceeded the horizontal extent of the associated

overflight period 'bar' (on both ends of the overflight bar), this would connote that the overflight was fully 'covered' by the relevant DIP clearance. This basic visualization protocol would be more closely analyzed and refined during the subsequent design activities in the second half of calendar 2004.

Multiple Levels of Referential Granularity

Our prior knowledge of TACC operations indicated that different roles within the mission planning / execution process path would need to invoke mission visualizations at multiple levels of referential granularity. For example, a mission planner would most likely need to invoke visualization of a single mission with which he / she was working at a given time. On the other hand, a supervisory staffer in the Execution Cell (e.g., a senior or duty officer) typically has to monitor a set of missions during his / her duty shift.

We knew from the beginning that it would be necessary to design the proposed timeline tool so as to support these distinct requirements and uses. This in turn suggested that whatever the timeline tool's visualization protocol and structure was to be, it needed to be coherent and consistent enough to be applied at both the individual mission and multiple mission levels of reference.

Knowledge Acquisition: July 2004

Knowledge acquisition (KA) in support of the proposed timeline tool concept was conducted during 12 - 14 July 2004 at TACC. This visit had originally been planned to occur in June 2004. However, scheduling problems resulted in our having to delay it for a month.

The WIDE team had laid out a KA plan intended to gather data on the following issues and themes:

- The 'horizontal' aspects of the TACC process path leading from mission planning through to execution (i.e., how peer roles jointly conducted this process)
- The data and information elements that accreted and fed forward during this process path
- The 'vertical' aspects of TACC operations (i.e., how supervisors / commanders such as DO's and Seniors maintained situation awareness and made decisions)
- The information requirements that pertain to such supervisory personnel
- Operational activities of certain roles we'd not previously examined in detail (e.g., barrelmaster, mission controller)

- Initial SME feedback on the notion of a timeline visualization aid

- The manner in which geo-spatial visualizations could support and augment timeline visualizations

- The manner in which geo-spatial and temporal visualizations could be correlated or organized to most constructively support TACC staff

During the three day July visit, the WIDE team interviewed and / or observed subject matter experts (SME's) representing multiple TACC roles or positions, including:

- Management / supervisory personnel from the barrelmaster unit (the people who assign specific aircraft to missions)
- Tanker mission planner
- Duty officer (DO)
- SAAM mission planner
- Senior
- Contingency mission planner
- Mission controller

In addition, we had meetings with TACC administrative and technical personnel, as well as with IT support contractors. The data collected from these interactions was supplemented with documentation collected during the KA visit, as well as relevant documentation forwarded to us during the remainder of July and August.

All four members of the WIDE 6.2 design team traveled to Illinois for this KA trip, and all participated in various subsets of the overall KA itinerary. In the wake of the KA trip, a summary report was assembled by Emilie Roth. The final edition of this report was distributed to the team, and we were prepared to undertake detailed timeline tool design work by 1 September. In addition to the data or knowledge base we'd accreted during the most recent KA exercise, we were also able to avail ourselves of data, experiences, knowledge, and documentation accrued during the previous AFRL WCSS projects with AMC (HISA, IFM, and GAMAT).

Design Team Procedures

The members of the WIDE design team were widely scattered in terms of geography. Dr. Roth was based in Massachusetts, Dr. Scott in Minnesota, and Dr. Whitaker / Ms. Thomas-Meyers in Ohio. Between the time of the July 2004 KA visit to Scott AFB and our return visit in December 2004, the team was never assembled in one place. As a result, the design team had to rely on distance communications and shared documentation to conduct our work. Our team discussions were conducted using group teleconferences (up to 3 per week), individual telephone conversations, and email messaging. The shared documentation consisted of (e.g.) Microsoft Word documents and Powerpoint slide sets.

Once we reached the point where we could start generating specific design specifications and concepts, Gina Thomas-Meyers and Randy Whitaker began a series of working sessions during October 2004. These meetings were the venue in which the specifics of the interface concepts were developed. As these particulars were generated, they were documented in the form of PowerPoint slides which were distributed to the team at large.

Starting in September 2004, the design team began a process leading from initial conceptualizations through progressive specification refinement to generation of design prototype specifications and a set of scenarios in which the prototype could be illustrated in relation to TACC operations. The following sections will outline the nature and the course of the activities comprising this process. Naturally, the design process was not as strictly 'linear' as the following discussion insinuates. Nonetheless, there was a generally unidirectional progression in our thinking and our products during the autumn of 2004.

In the interest of both brevity and clarity, the following sections will only summarize the 'high points' of the process conducted during autumn 2004. The illustrative summaries of both design work and design artifacts are distilled from a larger set of activities and documentation.

Laying Out our Design Objectives: September 2004

To document some of the team's tentative thoughts on a timeline tool, Ron Scott had drafted a summary position paper on the timeline tool concept and some of the factors that must be addressed in organizing our design work. This position paper was distributed to the design team in early September 2004. It stated the top-level design objectives as follows:

> "The purpose of the timeline tool is to display in one place many of the constraints that affect the success of a mission. Not all mission constraints can naturally be represented on a timeline view, but many can. The hope is that by presenting many constraints in one representation, by offering tools to easily do 'what-if's', by providing alerts when constraints are violated, and by providing the ability to pivot to other displays which can present some non-temporal constraints, we can develop a more robust capability to improve mission replanning and execution in the TACC."

This became the notional basis for framing our design objectives and our subsequent design concepts.

Organizing the Basic Elements to be Represented: September 2004

In the position paper cited above, Ron Scott had proposed a set of categories which might be applied to organize the types of data we needed to account for and interrelate in a timeline tool application. Summarily stated, these categories were:

- *Event* - an action that occurs at some defined point in the lifecycle of a mission.

- *Resource* - an item necessary to the successful operation of a mission.

- *Event Constraint* - a constraint asserting a relationship between two or more events.

- *Resource Constraint* - a constraint describing a limitation on the availability of a particular resource.

The initial design team deliberations used this taxonomy as the basis for determining a stable and constructive model for framing the subject matter to be represented. During the first month of the design activities we discussed and refined this model as an aid to organizing the features and factors we chose to portray in the timeline tool conceptual prototype. Strictly speaking this model is not explicitly depicted in the timeline tool prototype. This does not mean that it was abandoned or that it was irrelevant. The process of generating and refining this model contributed greatly to the design team's ability to address the subject matter in a coherent fashion. As such, even though the model does not represent a 'design product' per se, it served as an important 'design artifact' employed by the team during the generation of their design products.

Surveying Relevant Factors and Constraints: September 2004

By mid-September 2004, the design team was brainstorming to generate a list of candidate factors, constraints, and features which were relevant to both (a) temporally-correlated aspects of transport missions and (b) representation of such temporal aspects in a WCSS visualization. The initial list of items that this brainstorming generated was loosely organized with respect to two themes. The first was 'event' - i.e., a specific occurrence. The second theme was those events and / or constraints which related to one or another resource category. There was also a category of 'Other' for those items which we could not immediately subsume under either the 'event' category or correlate with a particular resource. An illustrative summary of the items generated in this inaugural brainstorming is given in Table 7.[6]

[6] There were actually a number of such listings generated during the course of this initial preparation and analysis work. The listing provided for illustration in the table is representative of the earlier versions with which we were working.

Table 7: Initial List of Events and Conditions Correlated with Resources

EVENTS	• Ground Events • Sequence of Events (SOE) • Takeoff / Landing • Air Refueling • Airdrop • Reaching a waypoint of a flight plan	• Crossing an airspace boundary (e.g., national borders, FIR's, and theaters of operations) • Scheduled maintenance events
AIRSPACE / PASSAGE	• DIP clearance constraints • Theatre clearance constraints (Theatre Slot Times) • Air Refueling track reservations • Availability windows for organized tracks • Temporal spacing constraints	• Communication zone constraints • Air traffic control regions • Weather • NOTAMS • Range • Divert opportunity regions • Intelligence
AIRCREW	• Type: basic, augmented • Qualifications/Certification • Availability • Crew duty day cycle • Crew scheduled return time	• Crew firm scheduled return time • Air Commander • Next/prior mission
AIRFIELDS / PORTS	• Operating hours (of ports) • Day/night periods • PPR time frame • Theatre slot times • Quiet hours • BASH hours • MOG (maximum-on-ground) timeframes	• On-site resources availability/accessibility • Take-off/landing factors (time-correlation) • ILS • Refueling capability (and fuel) • Material-handling capability • Crew accommodations
AIRCRAFT	• Type (including configuration, instrumentation) • Tail number • Availability timeframe(s) • Capabilities	• Maintenance status • Next scheduled maintenance • Aircraft 'turn-around' time on the ground (e.g., for loading / unloading)
LOAD / CARGO	• Type: Passenger, Cargo • If Cargo, type of Cargo (e.g., hazmat; human remains) • Availability timeframes	• Characteristics affecting aircraft feasibility and ops • Weight • Aircraft capacity status (e.g., full; empty)
OTHER	• NOTAMS • Weather conditions • Purpose of going to a destination	• Loads to be added / offloaded during multi-leg missions • Mission Type • Documentation

The set of categories and the sets of particular items would change as we further elaborated and refined our thoughts on the subject matter.

Setting the Scope of the Current Design Effort: September 2004

By the middle of September 2004, we had assembled a working summary of the planning factors and constraints generated in our brainstorming and discussions to date. This data set was developed into a structured table in which we categorized these elements into the following subsets:

- Events (estimated and / or actual times)
- Airspace / passage factors
- Aircrew factors
- Airfield / port factors
- Aircraft factors
- Load / cargo factors
- Mission CONOPS factors

For each element listed under one of these categories, the team specified whether that element should be (and / or could be) reflected in a timeline tool in the short term (within the next year). Elements that were judged incapable of implementation in the short term were flagged for deferral to a longer term. Elements whose relevance or implementation feasibility remained questionable were flagged for further investigation. An illustrative summary of how the team initially categorized and prioritized these features is provided in the series of tables (Tables 8-14) below.[7]

Table 8: Design Feature Evaluation: Events

Events: (Estimated and actual times)	Include in Short Term? Or ...?
Sequence of Events (SOE) for on-ground events	Long term
Takeoff	yes
Landing (including intermediate stops)	yes
Air Refueling / Rendezvous	Air refueling for short term
Airdrop	Yes (it's in GDSS)
Reaching a waypoint of a flight plan	Yes [for missions for which we can get a flight plan]
Crossing an airspace boundary (entry and exit points):	
Country border	yes
FIR	yes
ATC	investigate
Theater boundary	investigate
Reaching designated reporting time/place	Investigate
Scheduled airplane maintenance event (where and when can constrain availability of an airplane)	Investigate

[7] As was the case for the earlier illustrative table, this set of tables is a representative specimen drawn from what was actually a number of such listings that were generated and modified during the course of the design team's work.

Table 9: Design Feature Evaluation: Airspace / Passage

Airspace/Passage	Include in Short Term? Or ...?
Dip clearance constraints	yes
Theatre clearance constraints (Theatre Slot Times)	investigate
Air Refueling track reservations	yes
Availability windows for organized tracks	investigate
(Temporal) Spacing constraints (how close the planes can be)	investigate
Communication zone constraints	investigate
Air traffic control regions	investigate
Weather	yes
NOTAMs	yes
Range (given current load, fuel, weather)	investigate
Divert opportunity regions	investigate

Table 10: Design Feature Evaluation: Aircrew

Aircrew	Include in Short Term? Or ...?
Type: basic, augmented	Need to know, may not display directly on timeline
Qualifications/Certification	Investigate
Crew duty day cycle	yes
Crew scheduled return time (may members of a crew have different scheduled return times?)	yes
Crew firm scheduled return time	yes
Air Commander (may be specifically named in the DIPS)	Yes (may need to know, may not display directly on timeline)
Next/prior mission	Investigate (may go in an aircrew view)

Table 11: Design Feature Evaluation: Airfields / Ports

Airfields/Ports	Include in Short Term? Or ...?
Operating hours (of ports)	yes
Day/night	yes
PPR time frame	investigate
Quiet hours	investigate
BASH (Bird Aircraft Strike Hazard) hours	investigate
MOG timeframe:	investigate
On-site resources availability/accessibility	Long term
Take-off/landing factors (time-correlation)	Long term
ILS, other takeoff/landing related systems	Long term
Weather	yes
NOTAMs	yes

Table 12: Design Feature Evaluation: Aircraft

Aircraft	Include in Short Term? Or ...?
Type (including configuration, instrumentation)	yes
Tail number (needed for linking missions and scheduled maintenance, may be specified in DIPS)	yes
Previous/next missions	yes
Availability – when/how long?	Yes – 'higher level'
Capabilities (how much fuel can it carry, what altitudes can it fly at; security capabilities; countermeasures)	Long term
Maintenance status	investigate
Next scheduled maintenance	investigate
Hours remaining before mandatory grounding for maintenance (Phase Maintenance)	investigate
Communication capabilities	Long term
Aircraft 'turn-around' time on the ground:	yes

Table 13: Design Feature Evaluation: Load / Cargo

Load / Cargo	Include in Short Term? Or ...?
Type: Passenger, Cargo	yes
If Cargo, type of Cargo – e.g., hazmat level	At least some (e.g., hazmat)
Weight	No
DV/Banner/other high profile missions	Investigate – high priority
Characteristics that place restrictions with respect to type of plane that can carry it and special equipment needed:	Long term

Table 14: Design Feature Evaluation: Mission CONOPS

Mission Concept of Operations:	Include in Short Term? Or ...?
Mission Type: [including Operation it is flying in support of (e.g., Enduring Freedom) can impact what airspaces can fly over and what DIPS you need.]	May go into extended Sortie Palette
Connections among missions (e.g., refueling; one mission is going to 'save' or 'replace' a second mission)	Short-term Investigate

This structured summary became the focal data set for ongoing design team discussions. It also served as the initial specification for the types of information we wanted the timeline tool prototype to incorporate or address.

Translating Listed Factors into Visualization Elements: October 2004

During October 2004 the design team's focus shifted from enumerating the target subject matter to how that subject matter could or should be portrayed on the prospective timeline tool. This required a stricter accounting of the factors we'd identified in terms of:

- Their relevance to a temporally-based reference framework

- Any translation or permutation necessary to generate a temporal presentation of a given item

- How the 3 originally delineated elements (events, resources, and constraints) sorted themselves out in the context of temporal representation

- Relative groupings of the factors that would make sense to users as well as to us analysts / designers

In the following subsections we will summarize the actions taken to deal with these issues.

Sorting Out Events, Resources, and Constraints for Temporal Visualization Purposes

The early listings of relevant factors had agglomerated events, resources, and attendant constraints into one body of data. As can be seen in the illustrative tables above, all 3 types of items had been nominated and considered as things we might want our timeline tool design to accommodate. However, these constituted not just 3 distinct labels for things, but 3 distinct types of things. It was determined that we needed to more clearly specify what was being portrayed in the temporal visualization and how the set of 3 elements related to this target visualization.

Of the three elements, it was clearly the 'events' which were most intrinsically temporal. An 'event' is any occurrence, and must be associated with either (a) a particular point in time or (b) a specifiable period of time. Resources may have a period of time during which they exist or (e.g.) are available or viable. However, many of the things we included under the category of 'resources' (e.g., aircraft, airfields) are persistent in relation to the kinds of timeframes we anticipated portraying in a timeline tool. Phrased another way, resources were not so strictly bounded or 'punctuated' by time as events.

A running theme in our knowledge acquisition and our feature / factor enumeration efforts had been the 'constraints' impinging on TACC operations at both the individual and collective levels. Perceived constraints had served as some of the primary evidence for the types of events (and associated resources) we'd selected for consideration. Furthermore, we had identified the visualization of constraints and constraint conditions as a key payoff for a timeline WCSS. Allowing users to more readily and effectively identify and evaluate mission constraints was, we'd long believed, the key to improving TACC decision making processes. As such, 'constraints' had to be addressed. However, 'constraints' have a status much like that discussed for resources above. Although some situational constraints pop up during the timeframe of a given mission, others have a duration or persistence exceeding the timeframe intended to be portrayed on the timeline displays.

Our initial enumeration of candidate information elements to be included on a timeline display constituted a relatively loose-knit assemblage of events, resources, and constraints. These three distinct categories of elements could not be coherently portrayed on a single visualization. We needed a means for both prioritizing the key elements to be displayed and correlating these elements with relevant items from the other categories.

An Organizing Schema for Sorting Out Our Candidate Data Elements

After discussion of the defining characteristics of, and the interrelationships among, these three key concepts we laid out a more specific schema for organizing how we viewed the concepts' interplay in the context of temporal visualization. This model is illustrated in Figure 4.

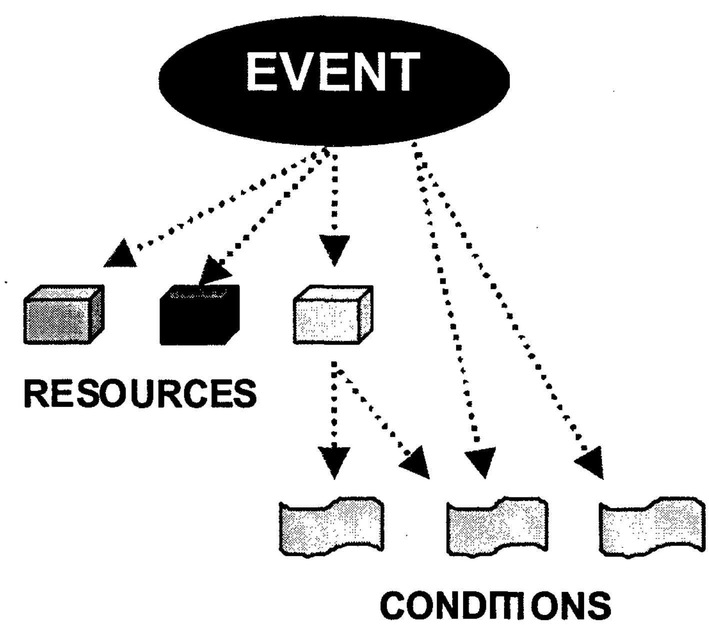


Figure 4: Organizing Schema for Events, Resources and Conditions

In this later-generation edition of our conceptual schema an event is taken to be a particular action or phenomenon having a specific time component. An event may be a single-point thing, or it may be something which has a duration that extends across an arbitrary span of the timeline representation. Such events were designated to be the focal elements portrayed on the timeline visualization.

Each event is related to or contingent upon one or more resources. In the context of this design effort for TACC, resources are specific items or elements associated with a mission or the context of that mission's execution. For example, a viable take-off event is predicated on having resources such as (e.g.) an aircraft, the designated cargo, a crew, a flight plan, etc. Types of such resources circumscribe the categories of elements to be portrayed on the timeline. However, the resources in and of themselves are not portrayed on the timeline. Only their 'temporal projections' (e.g., time of existence, applicability or

viability) is displayed. For example, the timeline would not illustrate a particular cargo per se. Instead, it would illustrate the period during which that cargo was associated with the given mission.

Both events and resources are associated with conditions - states or factors which affect the viability of resources or events comprising the mission. The reason that our prior allusions to 'constraints' got subsumed under the construct of 'conditions' was that there were identifiable states or factors whose effect on events and / or resources were not appropriately characterized as 'constraints'. For example, having a DIP clearance whose period of viability greatly exceeds the projected period of overflight (for the associated country) is an opportunity rather than a constraint. As was the case for resources, conditions are not directly portrayed on the timeline in and of themselves. Instead, the point at or period during which they pertain is shown. Sometimes the applicability of a condition is implicit rather than explicit - e.g., as when a gap between two timeline 'bars' indicates a problematical lack of coordination between any permutation of events and / or resources.

This organizational schema was not intended to be something shown on the timeline visualization. Instead, it was a working aid intended to be applied by the design team to sort out the various items and elements which our earlier brainstorming and collation exercises had generated. By applying this schema to the listings of relevant data elements and topics (cf. earlier discussion above) we were in a position to delineate a working set of timeline tool data specifications.

Conceptualizing and Defining 'Clusters'

As illustrated above, the design team had subdivided the candidate data items to be displayed into a set of categories. These categories were identified as sets or groupings which seemed to subsume chunks of the subject matter being nominated for representation. However, before concrete design specifications could be created it was still necessary to convert that loose set of topical categories into a set of specific elements comprising the timeline representations. There were multiple reasons why this translation needed to be done, as follows:

- *The organizing schema shifted the manner in which we were treating the subject matter.* The updated organization we'd applied to the subject matter put a priority on events as the key presentational elements, with resources and conditions having secondary roles. In effect, this meant that for anything to be guaranteed of portrayal on a timeline, it needed to be translated into or correlated with an event. To obtain consistency with the schema, some of the items on our earlier data element compilations needed to be re-characterized. As we did this, we found that some adjustments were needed with respect to the categories themselves.

- *The categories now needed to make sense from the perspective of a timeline artifact.* We had arrived at a point where we were beginning to outline the features of the timeline tool design concept. From this point onward we would have to contextualize our results with respect to the emerging product (the design concept). The way we'd categorized the subject matter earlier didn't uniformly or universally correlate well with particulars of an interface artifact. For example, the items we'd set aside in a general 'Other' category needed to be integrated with the other items in such a way as to comprise a clearly-delineated set of interface subdivisions.

- *The interface being designed became a factor in delineating data categories.* Subdivision of the data items in our early brainstorming and analysis exercises was based on the subject matter in and of itself. There were a considerable number of candidate data items to be included, and on-screen management of this data would be an issue. Furthermore, we wanted to create an interface motif which could be applied coherently at two different levels of referential granularity (individual missions and sets of missions). This meant that we would have to devise (e.g.) a subset of a full-blown individual mission representation to serve as that mission's summary depiction within the context of a multi-mission display. Such a subset or summary constituted a categorized set of data elements defined in relation to their role on the interface being designed, and not in relation to any subject matter taxonomic breakdown.

In the abstract, we could subdivide or classify our candidate data items in any number of ways. In practice, we needed to come up with a set of categories whose definition was as consistent with the WCSS product's functions as with the data. Because a single event may involve multiple different types of resources, we determined that distinctions among resources were the most useful bases for categorizing the relevant subject matter so as to be portrayed as events.

In the end, we arrived at a set of mainly resource-related categories as follows:

- *Geographic Elements* - Geo-spatial factors correlating with the mission such as nations overflown, departure and arrival airfields, control areas, etc. Representation of geographic items was necessary to correlate mission progress with locations. It may seem odd to consider 'geography' or 'location' as a resource (in the sense we first applied that label). However, the subsequent definition of 'resource' as any specific item associated with a mission or context of mission execution accommodates locations.

- *Aircrew Elements* - Factors relating the crew to the mission at hand, such as availability times, crew rest periods, and planned or mandatory return dates.

- *Aircraft Elements* - Factors relating the aircraft to the mission at hand, such as availability time, required maintenance periods, etc.

- *Airfield ('Port') Elements* - Factors affecting operating into and out of a given port, such as operating hours, closures, periods of maximum-on-ground (MOG) conditions, etc.

- *Ground Events* - Factors such as loading / unloading times, refueling times, etc. This category may seem anomalous, given the otherwise resource-orientation of the others. However, ground events are themselves contextualized with respect to temporal correspondence with the period during which a mission's progress intersects a given airfield (i.e., a resource).

- *Load / Cargo* - Factors relating the load to the mission at hand, such as arrival / availability schedules, periods during which certain key cargo types were on board, etc.

- *Permissions* - Factors reflecting administrative, legal, or operational permissions required for the conduct of a given mission, such as periods of diplomatic clearance (DIP) viability, etc. This category was largely motivated by the data we'd collected in our KA. For example, we were struck by the recurrent SME references to DIP issues as key sources of errors and re-planning demands.

- *Aerial Refueling (AR)* - Factors pertaining to an AR requirement, such as scheduled tanker rendezvous, etc. Aerial refueling was singled out for specific highlighting (as a category) owing to the perceived criticality of AR events reported time and again by our TACC SME's.

These categories (termed *clusters*) were to serve as the modular sub-frameworks for comprehensively representing mission factors relative to time. In effect, these were the categories derived from our initial notional taxonomies that were now to be treated as features of the interface tool being designed.

Specifying the Components in a Timeline Tool Display Suite: Autumn 2004

Formulation of our working set of 'clusters' afforded us the basic repertoire of visualization components to be included on a timeline display. The next step was to determine how this repertoire was to be employed in each of however many discrete timeline displays we planned to offer TACC users. We first needed to determine how many such displays were to be designed during this initial effort. From the beginning, we'd acknowledged two distinct levels of referential granularity evident in the work practices of our TACC clients:

- *Individual mission* - Throughout most of the process path leading from planning to execution, the various TACC positions address one mission at a

time. Additionally, when addressing one single mission the user is typically dealing with issues requiring reference to multiple data items or data types.

- *Multiple missions* - There are some roles - most particularly the supervisory roles of senior and duty officer - who routinely attend to more than one mission at a time. In such cases, the user typically wants to view summary data (especially top-level information on mission status) on each mission included in the set being viewed.

In the May 2004 TIM discussion of a proposed timeline tool, allowance was made for three options for timeline display reference: (1) a summary view over all pending missions; (2) a summary view over a selected subset of all pending missions; and (3) a detailed view of an individual mission. Options (1) and (2) are essentially two variations on a single (multiple mission) display capability, with the only difference being that in the case of option (1) the selection criterion is "all". As a result, we decided that a set of two timeline display tools - one for individual mission and one for a set of missions - would be sufficient to support our TACC users.

Specifying Position of the Timeline Tool within a TACC WCSS Suite: Autumn 2004

A related issue was the manner in which the timeline tool display(s) were supposed to be deployed and used relative to other WCSS tools supporting TACC operations. By the end of the GAMAT Phase II project, we had arrived at a point at which:

- We'd obtained at least rudimentary knowledge on the entirety of the TACC mission planning / execution process path and

- We could therefore think in terms of a general TACC-wide toolkit instead of one or another specific application.

By the time of the WIDE project and our timeline tool design efforts, we had already identified two pieces of such a general toolkit or WCSS suite. One was an application providing top-level situation awareness (SA) over the entire TACC workstream (set of mission 'cases' being processed at any given time). The second was the generalization of the GWM-WCSS into a 'Flight Visualization Tool' (FVT) which would afford a variety of TACC positions the ability to visualize flight routing.

Top-level situation awareness (SA) over the entirety of the pending mission workstream was not ignored. The AFRL WCSS team had repeatedly proposed some form of summary overview display allowing all TACC users to view all pending missions. This concept was first operationalized in the 'Sortie Palette' component of the GWM-WCSS during the GAMAT Phase II project. Such a top-level workstream overview capability can be ascribed to the Integrated Management Tool (IMT) display already in use within TACC. As such, we decided it was sufficient to presume existence of *some* form of top-level workstream oversight, regardless of the specific IT application providing it.

The Flight Visualization Tool (FVT) had been proposed (as something distinct from the GWM-WCSS) at the end of GAMAT Phase II (February 2004). A working GWM-WCSS application had been deployed at TACC. Given the fact that the proposed FVT was a variation on an extant application, we were comfortable presuming an FVT as a component of a general TACC WCSS suite and specifying a timeline tool deployment scheme based on its presence.

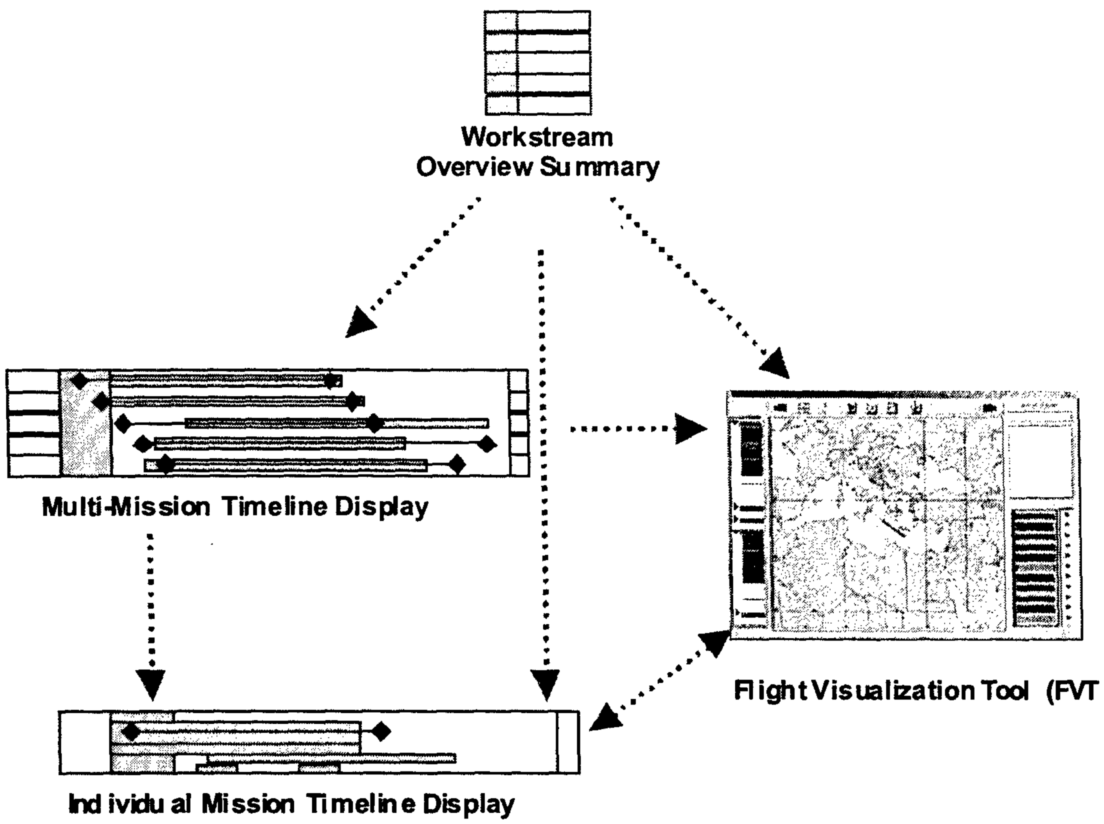


Figure 5: TACC WCSS Suite Concept

Our updated TACC WCSS suite deployment concept is illustrated in Figure 5. As the figure indicates, the top-level point of entry to the suite is the workstream summary. From a mission entry on that summary, the user was proposed to have a capability to invoke any of the other three suite components (either of the two timeline tool displays or the FVT). This would permit TACC users to invoke a visualization for either individual missions or collective sets of missions that was either framed with respect to 'time' (timeline tool) or with respect to 'space' (FVT).

Provision also needed to be made for 'toggling' between temporal and spatial visualizations for the same mission or set of missions. This would permit TACC users to analyze a given situation using both visualization modalities should the need arise. For example, if someone in the Execution Cell needed to assess possible divert airfields he / she would thereby be enabled to evaluate options in terms of either time (using the timeline tool) or distance (using the FVT).

Designing for Active Decision Support: October - November 2004

The timeline tool (like the previous WCSS design concepts) was intended to serve not only as a passive visualization aid, but also as a dynamic (re-)planning tool. To allow users to employ the timeline tool in this manner, we had to specify a use concept that provided for manipulation of the temporal elements displayed, followed by updated display of the ramifications of any changes thus made. The most troublesome issue in laying out such a strategy was ensuring that users did not lose sight of the distinction between (a) the actual recorded state of a mission 'as is' versus (b) a prospective state of the mission resulting from 'what-if' simulation / manipulation actions.

We decided that the most practical solution would be to allow for two 'modes' of visualization for a particular mission:

- *'Visualization' Mode* - a mode in which the user is viewing the most current 'as is' data for the given mission.

- *'Simulation' Mode* - a mode in which the user has a local copy of the current 'as is' data which can be dynamically manipulated to create and evaluate 'what-if' variations on the mission data.[8]

Our inaugural timeline WCSS design concept calls for two variations on the individual mission display. The first (the default view) is the 'visualization mode'. If the user wants or needs to invoke 'simulation mode' for analysis or (re-)planning purposes, he / she would be provided the ability to invoke a separate 'cloned' individual mission display which would be subject to manipulation. By providing two distinct (and potentially simultaneously on-screen) interfaces, we could:

- Maximally reinforce the distinction between the two modes of addressing the subject matter.

- Minimize confusions that could arise from forcing the user to constantly recall which mode he / she was in.

- Allow for certain variations in interface features - some of which were peculiar to one or the other mode.

[8] The term 'simulation mode' has its shortcomings, but we have not yet found a better candidate label. There were suggestions that we might well label this a 'predictive' or 'projective' mode. The one drawback to those alternatives is that we were making allowance for 'what-if' simulation capabilities that weren't limited to future / hypothetical events. The reason we allowed for modeling and manipulating past events was that such a capability might be useful for (e.g.) quality / performance analyses and / or training purposes.

Specifying a Timeline Tool Display Layout: October - November 2004

The first thing we had to do in laying out the timeline tool design specifications was to determine what data would be encapsulated in the summary visualization which would serve as the common element in both the individual and multi-mission displays[9]. To provide a summary point of reference for users, we then devised a 'core' set of data which would serve as both (a) the 'short-form' summary timeline representation of a given mission and (b) the focal component within a presentation of multiple clusters.

This second function (within an individual display) was itself derived as an exercise in work-centered design. Because the individual mission display (in full form) would incorporate all the clusters we'd devised - each one of which might have multiple subsidiary data components - we felt it necessary to provide a sort of summary 'header'. This header would serve as a summary point of reference just as it (in isolation) served as a mission summary in the context of the multi-mission presentation.

In the following subsections we shall introduce and review the visual components of our timeline tool design concepts. This will only be a cursory overview. For further details, the reader should refer to the WIDE 6.3 final report.

Procedure for Developing the Timeline WCSS Design Concepts

The groundwork laid earlier - e.g., the generation and refinement of a set of factors that could be portrayed in temporal terms - provided a strong foundation on which to base the designs. The process of finding a good 'mix' of design elements in accordance with perceived user requirements was still a challenge. In the beginning, a more or less 'top-down' approach was used to generally sort out what should and / or could be portrayed on a timeline tool. There were diverse facets to this sorting problem. We needed to ascertain 'where' something should be presented, 'how' its presentation should be configured, 'how much' information would have to be conveyed in the presentation, and 'what' options should be implemented for (e.g.) cueing the user with respect to decision-critical states and conditions.

In the end, the process of generating design specifications was a labor-intensive activity which frankly involved a lot of 'trial and error'. Tentative design elements proved to be unwieldy or impractical once applied to the next step in a design construction. Some features recommended themselves in terms of adding informative aspects for the user, but clashed with other (e.g., procedural) aspects of how the WCSS might be employed. In summary, the process was not strictly linear, though it moved forward in a steady

[9] Different terminology was employed by different people at different times for these two main types of visualizations. For example, the 'individual mission display' was occasionally called a 'single mission display'. Similarly, during autumn 2004 and the December 2004 Design Review at AMC we mainly referred to the multi-mission display as a 'composite display'.

progression. There was a lot of debate, negotiation, and re-negotiation of design motifs, themes, features, and details. The bulk of the detailed design work was conducted by Gina Thomas-Meyers and Randy Whitaker in a series of sessions during October and November 2004.

Because we were operating on the presumption that a multi-mission timeline display would be a composite set of visualization elements replicated in the individual timeline display, we concentrated on the individual mission case first. Another reason for focusing on the individual mission display was that it was at this level of granularity that both the widest range and deepest levels of details had to be identified, sorted through, and worked out to produce viable design concepts.

The 'Core' Visual Element

Based on our analyses, we chose a limited set of features to be portrayed in the summary 'Core'. These included a general timeline of events, aerial refueling (AR) timeframes, projected time in air, geo-spatial areas being overflown, and diplomatic clearances (DIP's) associated with these overflown areas. The inclusion of AR and DIP data was based on the priority assigned to these topics by users as foci of attention and sources of problems. In other words, our prioritization of these elements corresponded to their importance from the user's first-person perspective - a key theme in work-centered design. The basic form of the Core display is illustrated in Figure 6.

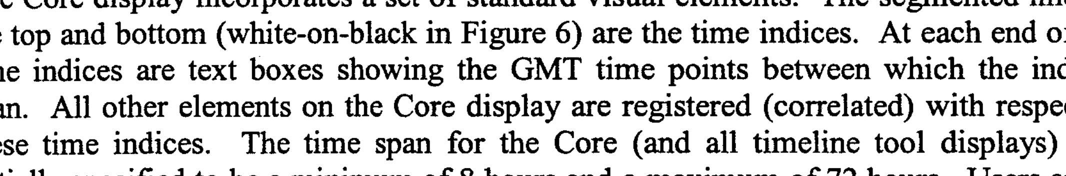


Figure 6: General Layout of the 'Core'

The Core display incorporates a set of standard visual elements. The segmented lines at the top and bottom (white-on-black in Figure 6) are the time indices. At each end of the time indices are text boxes showing the GMT time points between which the indices span. All other elements on the Core display are registered (correlated) with respect to these time indices. The time span for the Core (and all timeline tool displays) was initially specified to be a minimum of 8 hours and a maximum of 72 hours. Users are to be allowed to 'zoom' in and out in increments of 8 hours.

The upper portion of the Core contains a set of visual elements illustrating the state of the mission or sortie being viewed. The general visualization protocol for the graphical aspects of this Core component is illustrated in Figure 7.

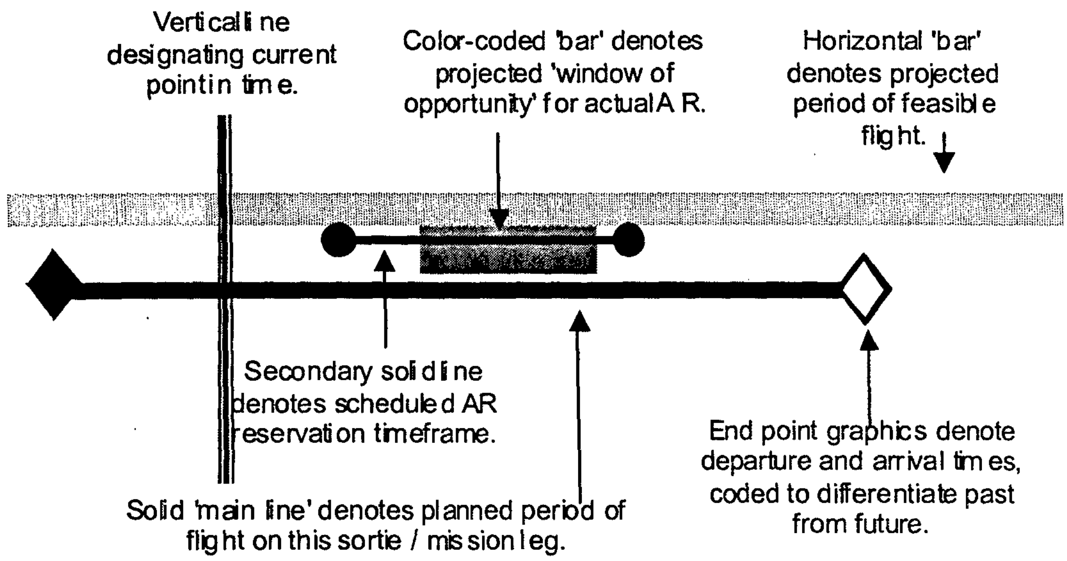


Figure 7: Basic Visual Elements in the Core Display

The solid 'main line' denotes the period during which the sortie is in progress. A vertical 'current time indicator' cues the user as to where the past ends and the future begins. Above the sortie 'main line' are two parallel visual elements - one for aerial refueling (AR) and one for 'time in air' (projected period of feasible flight). The 'time-in-air' projection cues the user on the temporal boundary for making flight changes by illustrating when the aircraft will probably have to cease its current flight. Two distinct graphical elements (a line and a color-coded 'bar') cue the user on the planned AR timeframe as well as the best-projected 'window of opportunity' for feasible AR.

To either side of this graphical sortie data summary are 'tabs'. These tabs provide an area which can be color-coded to cue users on alert status. As was the case in our prior WCSS designs, we used a 'stoplight metaphor' allowing for red (problem), yellow (caution), and green (OK) indications. Within these tab areas are text boxes. These text boxes are to contain airfield identifiers (ICAO codes) and time entries. Time entries will be provided for both planned and actual / projected values. Using this combination of features, a user can readily ascertain *where* a sortie begins and ends, *when* it begins and ends, and whether it is proceeding according to planned itinerary.

For multi-leg missions, the graphical protocol was extended to include the following elements:

- Dotted 'main line' portrayal for time on ground at a given port
- Text labeling to denote which leg (e.g., '2 of 3') was being represented

The lower portion of the Core portrays national overflights and DIP coverage for periods of such overflights. This portion of the Core is illustrated in Figure 8.

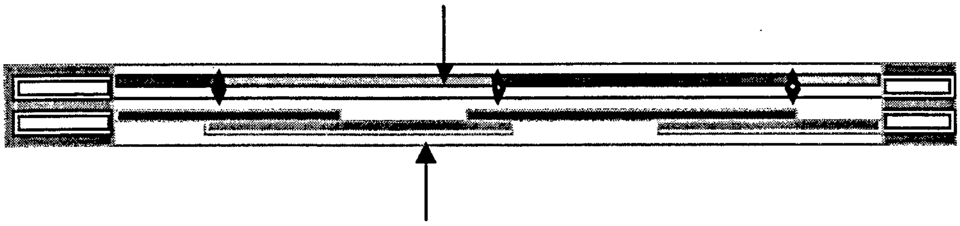


Figure 8: Overflight / DIP Elements in Lower Core

The upper row of this Core portion contains indicators for the nations being traversed during the given mission. The nations are designated as partitions of a single line, because you can't be in 2 places at once. International transitions are coded with diamond waypoint indicators, to aid the user in visualizing border crossing times. Periods of DIP coverage are coded with horizontal 'bars' in the lowermost section. These 'DIP bars' are staggered to allow for overlaps. At either end (left / right) of these elements are tabs with text boxes cueing the user to the type of data portrayed in the associated subsection. These tabs are independently capable of being color-coded for alert status as appropriate.

Both national overflight period and DIP indicators are intended to be coded in accordance with the 'stoplight' coding metaphor mentioned earlier. This permits us to flag mission status and problems with respect to access to a given airspace (whether or not it's DIP-related), as well as flagging status or problems with respect to DIP's themselves. Any period of 'non-viable' overflight is to be coded red, while any period of 'viable' flight (if any) continues to be coded green (or yellow, if an intermediate 'caution' status is applicable). Specific DIP clearance involved in a fault condition or constraint violation is coded red. This coding scheme gives the user direct situation awareness on what portion of mission is in jeopardy with respect to general geo-spatial correlates.

The modularity of 'summary / core' versus 'detailed cluster' representations afforded us the ability to meet two distinct needs among the target users. A set of 'core' summaries could be presented to give situation awareness over multiple missions (something needed by supervisory staff and execution phase monitors). A complete set of clusters for a given mission would be most useful for personnel focused on (re-)planning or analyzing one particular mission. In the following sections the inaugural set of timeline tool clusters will be briefly introduced.

Geographical Visualization Cluster

The 'geographical' cluster displays temporal data correlated with particular geo-spatial items. As mentioned above, the nations being overflown are portrayed in the Core section owing to their general importance to situation awareness and decision making. In

the geographical cluster (Figure 9) there are four subsections illustrating timeframes for: time in a given theater of operations, any applicable weather (WX) watch areas intersected by the sortie, periods on organized tracks, and periods during which the mission is operating within a given FIR (flight information region).

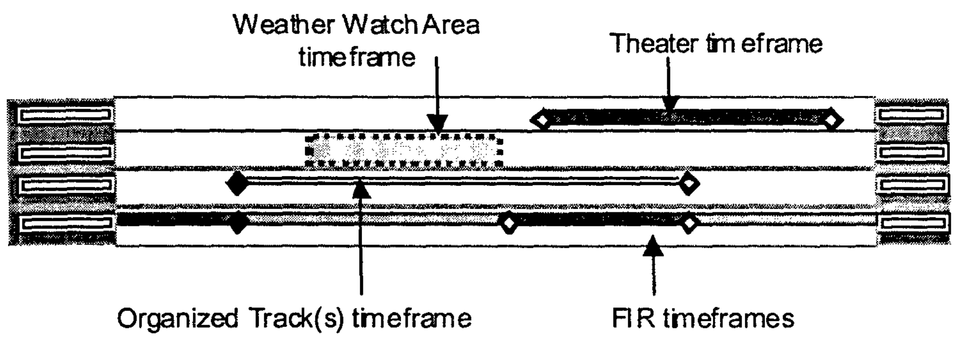


Figure 9: Geographical Cluster

The basic visualization protocols outlined earlier pertain to this cluster. FIR traversal is coded in the same manner as national airspace traversal - as a series of partitions in a single horizontal element. WX watch areas are portrayed as 'bars' rather than lines to highlight their presence, and they are presumed to be coded either yellow or red to indicate a cautionary status.

Aircrew Cluster

The aircrew cluster depicts data concerning the availability of the aircrew assigned to a given mission, as well as illustration of the timeframes subject to crew duty day and rest period constraints. This cluster consists of 3 'layers' organized from top to bottom in accordance with these topics. The aircrew cluster is illustrated in Figure 10.

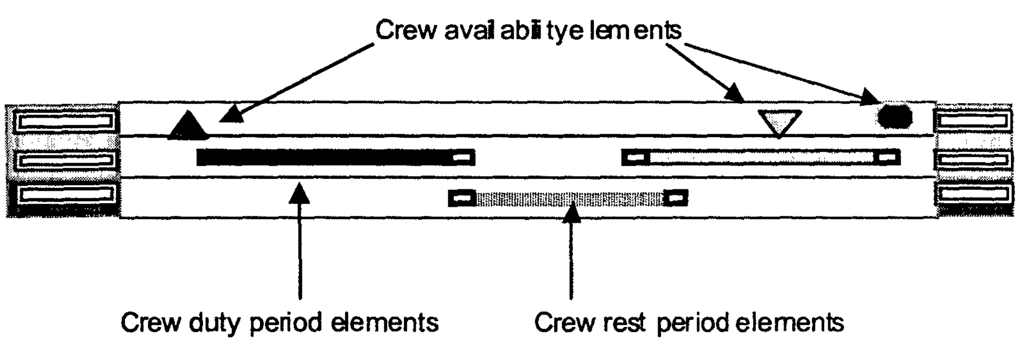


Figure 10: Aircrew Cluster

The crew availability component uses triangles to denote the point at which the crew is available for the mission (upward-pointing triangle). Crew return times are also illustrated. The (yellow / cautionary) downward-pointing triangle denotes the crew scheduled return time, and a red octagon indicates the crew's firm return time. These

designators cue the user to the absolute temporal boundaries during which the crew is nominally available as a mission resource. Crew duty and rest periods (in the lower 2 layers) are denoted with color-coded 'bars'.

Airfield (Port) Cluster

The airfield cluster (Figure 11) depicts data concerning the time a mission is at a given airfield or 'port'. The data contained in the airfield cluster is localized (horizontally) to that portion of the horizontal timeline during which the mission is present at the given airfield. This period is coded with background shading to aid the user in distinguishing on-ground periods from flight periods. There are 5 layers initially designated for inclusion in the airfield cluster. From top to bottom in our original design concept, these layers are associated with airfield operating hours (ops hours), light / dark periods (i.e., day and night), quiet hours (as applicable), BASH (bird strike / hazard) periods, and maximum-on-ground (MOG) periods (as applicable).

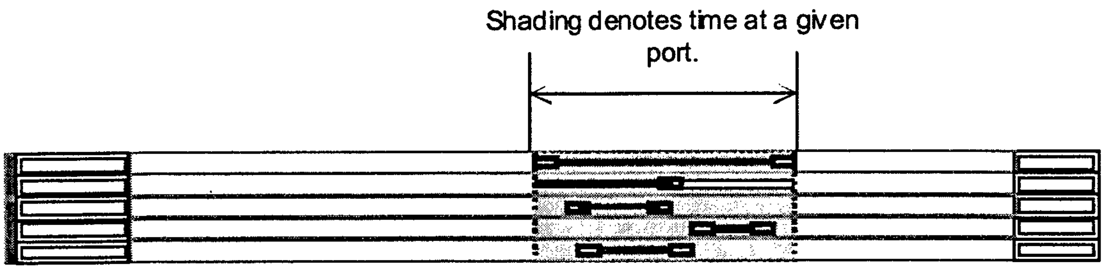


Figure 11: Airfield Cluster

Within each of the 5 layers, periods reflecting the associated state, condition, or phenomenon are depicted by horizontal 'bars'. The day / night 'bar' is partitioned (as appropriate) into black and white segments to indicate day and night conditions. The other four are capable of color-coding to indicate relative alert status.

Aircraft Cluster

The aircraft cluster depicts data concerning the availability of a given tail number for the mission at hand. This is the extent of the aircraft data we believed we could provide in a first-generation timeline tool demonstration prototype. The general form of the aircraft cluster is illustrated in Figure 12.

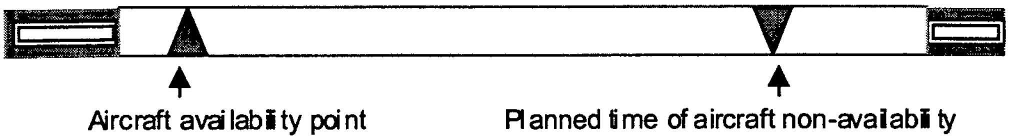


Figure 12: Aircraft Cluster

Triangular pointers cue the user on when the aircraft is available as a mission resource and when it is planned to no longer be available. These designators are color-coded to afford a means for cueing the user should either of these time points become problematical.

Ground Events Cluster

The ground events cluster (Figure 13) depicts data concerning the timeframes for functions and activities performed while the aircraft is on ground at a given airfield or port.

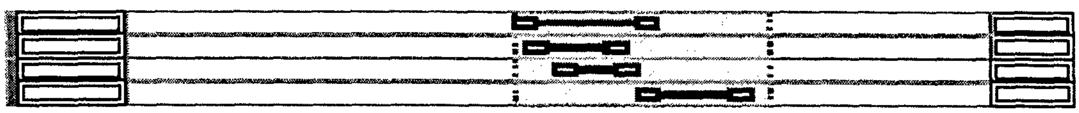


Figure 13: Ground Events Cluster

In its initial version, we have allocated four types of ground event data to be illustrated on this cluster. Each type has its own 'layer' in the display. These layers, from top to bottom, are associated with minimum standard preparation time (for takeoff), crew exchange time (if applicable), standard or projected offloading time, and standard or projected onloading time. We included these data types in the first-generation specification because we understood them all to be specifiable from existing information sources.

Additional activities which we had considered included standard times for refueling the aircraft, times for de-icing (as applicable), and times for taxiing and parking. We omitted these latter elements from the inaugural ground events cluster on the basis of our inability to ensure the relevant data was available and accessible for employment in a first-generation demonstration prototype.

Load / Cargo Cluster

The load or cargo cluster was designed to provide ready cueing on presence (and duration of presence) of those categories of cargo / load known to impose special requirements / constraints. One of the things we'd learned over the years of studying AMC / TACC was that the interactions between certain load categories and administrative or legal requirements could be both tricky to manage and critical to executing a mission as planned. The general form of the load / cargo cluster is illustrated in Figure 14.

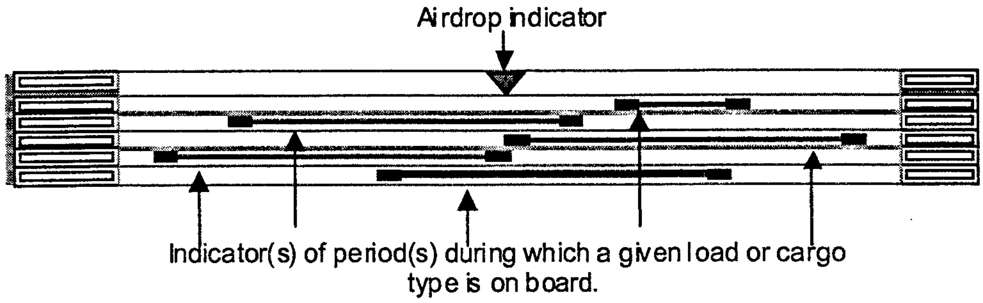


Figure 14: Load / Cargo Cluster

The inaugural version of the load / cargo cluster includes a total of 6 'layers', each associated with a given event or load category. The topmost layer is associated with airdrops. A triangular graphic is used to designate the point(s) at which a cargo is to be airdropped. Airdrops occur in a minority of TACC missions. However, when they are conducted they can be extremely time-critical (as well as mission-critical). The next 5 layers are each associated with a given load type. In the initial design concept, these are organized from top to bottom as follows: passengers (Pax), hazmat, human remains, medical evacuation, and nuclear.

Additional load / cargo conditions that we identified as relevant to mission decision making included periods during which the aircraft is traveling empty, periods during which airlift capacity is available (i.e., the aircraft is only partially laden), and periods during which specially-required load handling equipment is on board. These additional categories were not included in the inaugural cluster design because we could not ensure the relevant information was available or accessible at this time.

Aerial Refueling (AR) Cluster

Our inventory of candidate clusters had always made provision for portraying AR information. Our working set of cluster allocations had included a separate AR cluster. As mentioned earlier, we decided that the criticality of AR information (to mission success) was sufficient to warrant embedding AR data in the Core. As such, the inaugural timeline tool design concept does not incorporate a separate AR cluster.

Permissions Cluster

Transport missions must often be conducted in areas which require certain permissions to be granted. The most obvious such permission is a diplomatic clearance (DIP). Another common example is a prior permission request (PPR) mandated for being allowed to use an airfield administered by (e.g.) another DOD service. As discussed earlier, temporal information related to DIP clearances was judged to be sufficiently critical to warrant incorporating it in the Core.

This left PPR's as the sole permission type believed to be capable of visualization in the initial timeline tool prototype. In the initial design concept specifications, PPR's are to be portrayed as color-coded bars as applicable. This is illustrated in Figure 15.

Figure 15: Permissions Cluster

A third permission type - theater clearances - has been identified as an item we would like to include in the timeline display. However, we did not make explicit allowance for theater clearances in the inaugural design specifications on the grounds that we could not ensure the relevant data was available or accessible.

Assembling the Core and Clusters into an Individual Mission Timeline Display

Once we'd laid out specifications for the Core and the various clusters, the next step was to bring these conceptual pieces together in a single interface concept. Following the principles developed in earlier WCSS projects, we assembled these elements into a centrally-positioned visualization, around which peripheral features (e.g., for controls and navigation) were to be positioned. We elected to place the Core element at the top of the central visualization. The clusters were to be layered beneath the Core within this central visualization area, with the 'current time' indicator and time index components providing visual means for correlating these diverse elements with respect to each other. The layout of the individual mission timeline display - in default 'visualization mode' - is illustrated in Figure 16. This figure indicates the manner in which the Core and cluster components were assembled to comprise the full WCSS display.

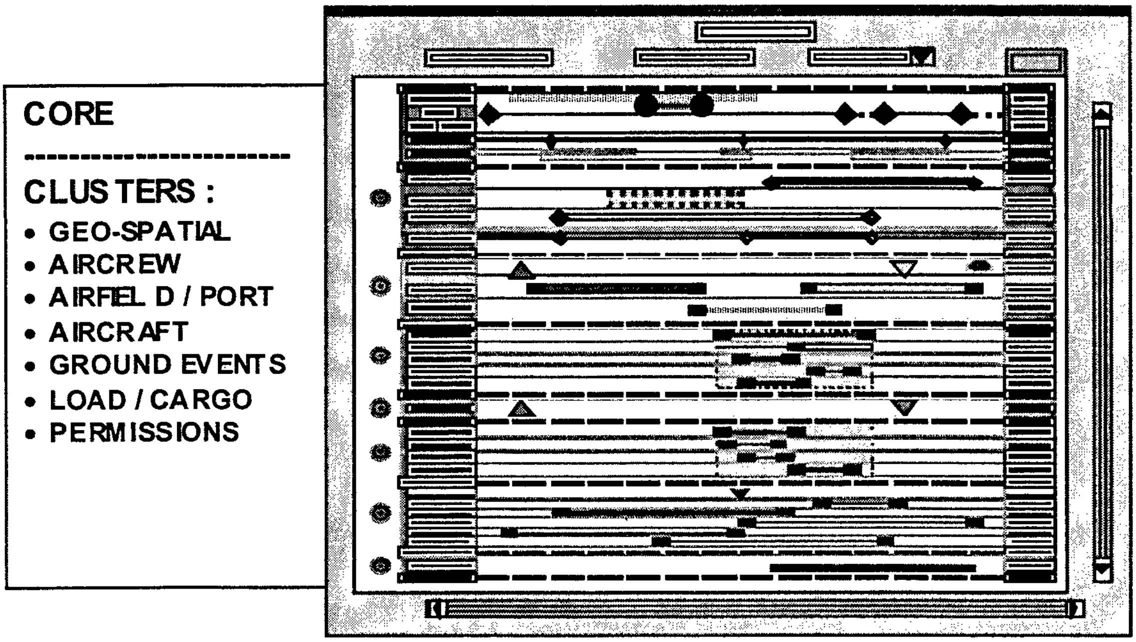


Figure 16: Overview of the Individual Mission Timeline Display Concept
('Visualization Mode' Illustrated)

Specifying the Peripheral Elements Completing the Individual Mission Display

As mentioned above, the individual mission concept contained *both* a central visualization comprised of the display elements discussed earlier as well as peripheral elements for control and navigation purposes. This meant that we had to specify the set of peripheral features that would be necessary to complete the individual mission display concept. For the purposes of this first edition of the timeline design work, we included the following additional elements in the default 'visualization mode' display:

- *Vertical Scrollbar* - We added a vertical scrollbar (illustrated to the right of the central display in Figure WC-14). This was intended to afford the user the ability to scroll up and down across what could in some cases prove to be a long and elaborate vertical 'stack' of clustered data.

- *Horizontal Scrollbar* - We also added a horizontal scrollbar (illustrated below the central display in Figure WC-14). This was needed to permit the user to scroll right and left to check mission features occurring earlier and / or later than the timeframe presented in accordance with default preferences.

- *Activation Buttons for the Cluster* - We added a series of radio buttons (illustrated to the left of the central cluster display in Figure WC-14) to give users the ability to toggle individual clusters 'on' and 'off' as they saw fit. This feature was proposed to allow users to selectively invoke or maintain on-screen those clusters most pertinent to their current task.

- *Mission Identifier* - Along the top of the interface unit we placed a series of elements to provide a 'header'. This included a row of 4 elements immediately above the central display area. For the leftmost position in this row we proposed a text box in which the mission number of the currently-focal mission or 'case' would be persistently displayed. This would provide continuous cueing on which of possibly several cases the display represented.

- *Current Time* - The second-from-left element in the row is a 'current time' text box showing the user the GMT time associated with the 'current time' vertical bar on the central display area.

- *Zoom Level* - The third-from-left element in the row is a drop-down menu (with persistent display of the current selection). This drop-down menu contains the available levels of temporal granularity the user may select for his / her visualization. Our initial design specifications allowed for a range of temporal 'zoom levels' from 8 up to 72 hours, in increments of 8 hours.

- *Simulation Mode Button* - The rightmost element in the row is a button permitting the user to invoke a 'simulation mode' display which can be actively manipulated to generate and evaluate 'what-if' conditions pertaining to the current mission.

- *Mode Reminder* - At the very top of the display (above the row of elements just described) is a text box which prominently displays the fact that the user is in 'visualization' mode. This redundant cueing was included to help minimize any SA errors when multiple displays are on-screen.

Defining the Form of the Individual Mission Timeline Display's 'Simulation Mode'

As discussed earlier, the individual mission timeline display was to be available in two 'modes', of which the 'visualization mode' was to be the default. The other mode was to be a 'simulation mode' in which the user could actively manipulate the display elements to denote variant conditions, analyze the ramifications of such tentative alternatives, and document them for re-planning or other purposes. The general format for the simulation mode version of the individual mission display is illustrated in Figure 17.

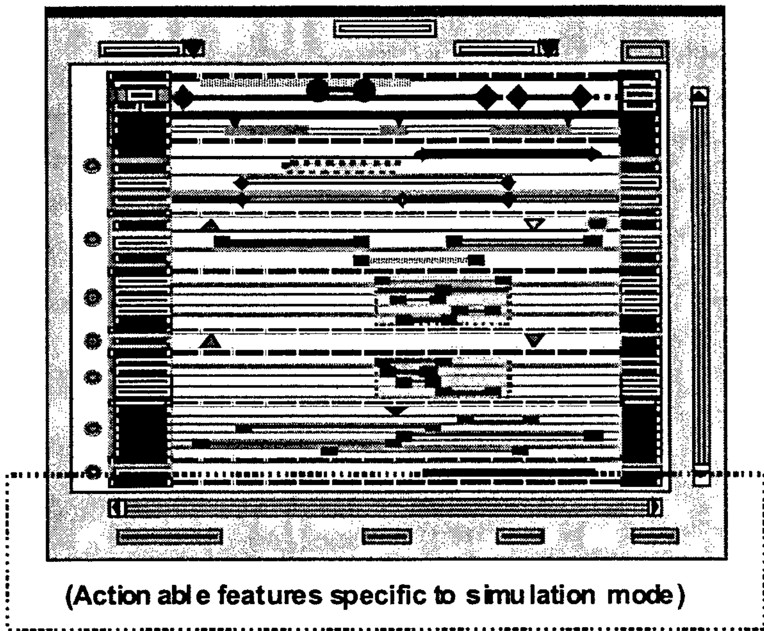


Figure 17: Overview of Individual Mission Display - Simulation Mode

For the most part, the simulation mode edition of the individual mission timeline display is identical with the visualization mode edition. The Core and clusters structure for the central visualization area is the same, as are the scrollbars and activation buttons. The header area is largely the same, in that it includes the mission identifier text box and the zoom level controls. No 'current time' cue is given, because in simulation mode the user is operating outside the fixed timeframe of the 'real world'. The mode toggle button that triggered the simulation mode (on the visualization mode default edition) is mirrored on the simulation mode edition by a button which toggles back to (invokes and brings to the foreground) the visualization mode display for the same mission.

In the simulation mode display, direct manipulation of the graphical elements in the central visualization area is permitted. For example, the user can 'click and drag'

The most significant layout difference is that we added a set of 'actionable features' in the form of a set of buttons along the lower periphery of the display palette. Each of these buttons is a 'trigger' that enables a specific action relative to the state of the simulation mode display. From left to right along their bottom row, these buttons are defined as follows:

- *Automated Processing Trigger* - The leftmost button in this row triggers an automated update and analysis of a new mission state (vis a vis the timeline presentation) as specified by the user through direct manipulation. The concept is for supporting agents to process a local copy of the mission data to generate a simulation state reflecting the user modifications (without having to modify the organization's actual data assets).

- *Revert to 'Last-Saved' Trigger* - The button second from left is intended to let the user revert to the most recently saved state of the simulation. This is to permit a limited form of immediate 'backing up' (a la a Web browser) when generating a new simulated set of conditions.

- *'Save' Trigger* - The button second from the right is intended to let the user save a local file containing the current state of the simulation display. This is intended to permit the user to document candidate (re-)planning states of affairs for future reference or to use as supporting documentation in a request to another position or unit.

- *'Print' Trigger* - The rightmost button allows the user to send the current state of the display to a printer to generate hardcopy documentation.

Defining the Form of the Multi-Mission Timeline Display

The multi-mission timeline display is the one that is designed to allow a variety of roles to obtain summary situation awareness over a selected set of missions. It is essentially an ordered set of Core representations drawn from each of the missions selected for inclusion. The form of the multi-mission display follows the general layout applied to the individual mission timeline displays. However, a set of peripheral features - many of which are peculiar to the multi-mission display - have been included in the inaugural edition of the design concept. A summary illustration of the multi-mission display layout - highlighting these peripheral features - is given in Figure 18.

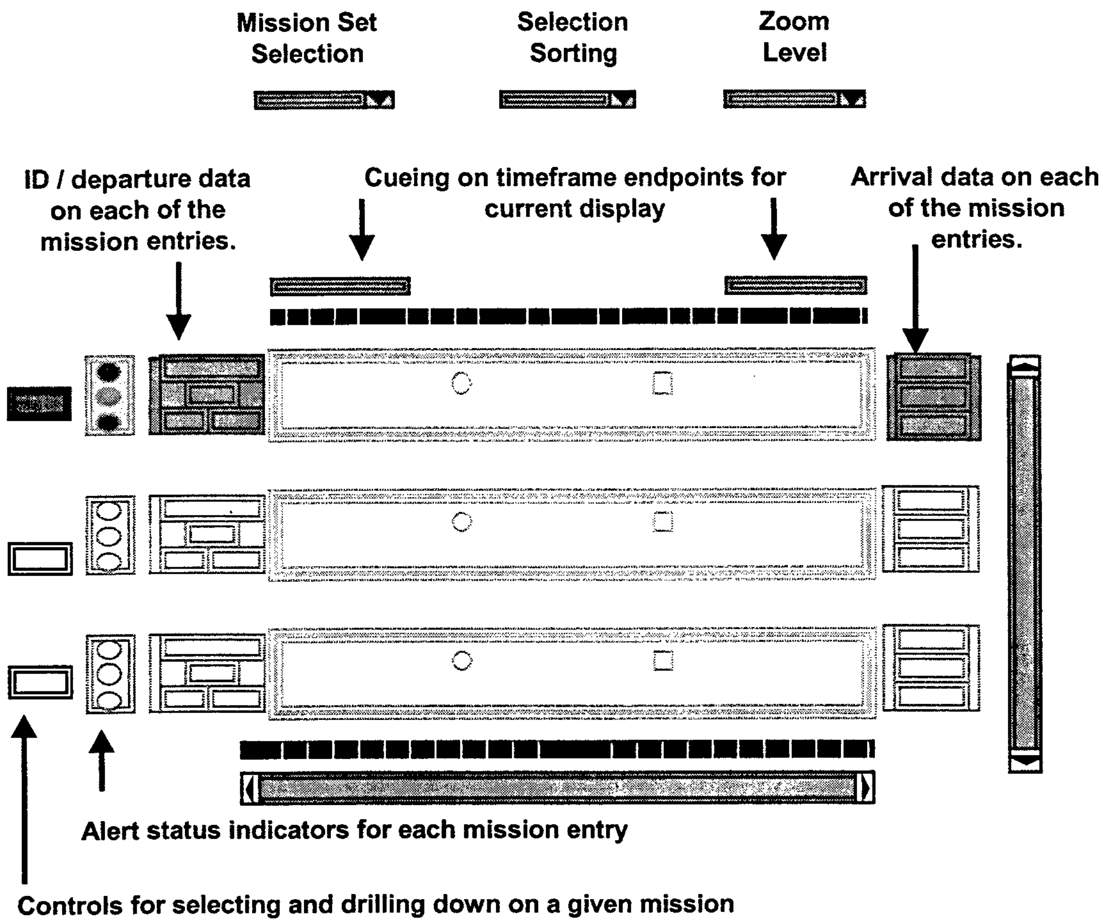


Figure 18: Overview of the Multi-Mission Timeline Display Concept

The multi-mission display's peripheral features include the following (cf. Figure 18):

- *Mission Set Selection* - A drop-down menu on the left in the upper row (above the central visualization) provides the user with the ability to select a set of missions for display.

- *Selection Sorting* - The center drop-down menu in the uppermost row provides the user with a set of criteria upon which he / she can sort the set of missions being displayed.

- *Zoom Level* - The rightmost drop-down menu in the uppermost row provides the user with the same options for 'zoom' (visualization timescale granularity) as previously described with regard to the individual mission timeline display.

- *Selection and Drilldown Buttons* - The buttons to the extreme left of each Core visual element serve to allow the user to select a given mission from the displayed set and trigger the presentation of an individual mission timeline display. This is the mechanism by which the user is intended to be able to

'drilldown' on a given mission as needed. Such a drilldown maneuver invokes a separate palette for the individual mission display.

- *Alert Status Indicators* - The graphical elements immediately to the left of each Core element are intended to cue the user to the top-level / summary alert status for the associated mission. As in other instances described earlier, the design concept calls for the 3-way 'stoplight metaphor' of red / yellow / green color coding for 'problem' / 'caution' / 'OK' conditions, respectively.[10]

Making the Case for the Timeline Tool WCSS

Our work-centered orientation had given us the basis for understanding and analyzing the actual domain in which work subject matter and work activities interact. By focusing on the work and the actual workers, we were able to maintain a focus on real problems affecting actual operations. By the time we had concluded our problem analysis and conceptual design work, we had generated a coherent intervention strategy interrelating technical innovations with operational and functional payoffs, as outlined in Table 15.

Table 15: Summary of Intended Timeline Tool Payoffs

FUNCTIONAL ENHANCEMENTS *Payoffs relating to individual performance*	Better inform TACC staffers via: • Fused visualization of disparate mission elements' interrelationships • Ability to perform 'what if' simulations to support decision making. Enable TACC staffers to: • More effectively plan and monitor missions • More effectively recognize and respond to mission problems
OPERATIONAL ENHANCEMENTS *Payoffs relating to team and organizational performance*	• Provide better situation awareness (SA) on mission events and hence on mission viability • Make interactions and constraints associated with events visible • Provide mission 'timeline' visualization
TECHNICAL INNOVATIONS *The means employed or created to achieve the payoffs*	• Fusion of all relevant data and correlation with a 'time context' • Ability to address and manipulate temporal data relating to different events and phenomena • Coherent linear 'timeline' schema into which relevant data on (e.g.) events can be mapped • Access to the varied data / database resources within TACC (e.g. Schedule, Route and Resource data) • Automated support (agents) to evaluate mission parameters and cue users on any problematical states, constraints, etc.

Table 15 is based on the presentation we made to our TACC clients during the Design Review in mid-December 2004. It summarizes the main points in our case for moving forward with timeline tool development and testing. This case was framed with respect to a set of 3 'layers', each dependent on the one(s) beneath. Technical innovations

[10] As of the date this final report was being drafted, we were still staying with depiction of the 'full stoplight' for this element (i.e., coding one out of three displayed subelements) so as to reinforce the color coding with positional cueing.

bringing together more effective data fusion and display were claimed to facilitate better user understanding of mission subject matter and hence facilitate more efficient and effective decision making.

Illustrating the Timeline Design and Use Concepts to our TACC Clients

The debut of the timeline tool design concepts in front of a TACC client audience occurred in early December 2004, at a series of design review meetings. The design concepts were introduced in a pair of PowerPoint presentations (Whitaker, 2004) - the first of which gave the intervention rationale derived from our KA and analysis, and the second of which stepped through the details of the design concepts.

Over the years, our AFRL WCSS team has learned that it is most effective to demonstrate a new WCSS design's use concept using a scenario or 'vignette' based on actual work practices and situations. During October and November 2004, the design team generated a set of such scenarios for this purpose. Documentation of this scenario set is provided in Appendix A.

Obtaining Feedback on the Timeline Tool Design Concept from our TACC Clients

During our December 2004 design review trip to Scott AFB, the design team conducted a series of interviews with TACC subject matter experts. In each of the interviews the SME's were questioned about data access and visualization issues. They were shown a small summary set of timeline tool illustrations derived from the materials used in the formal presentations. The SME's were given an overview of the design features and a brief introduction to the use concept. Comments and feedback were recorded for future reference.

Ongoing WCSS Design Issues

The design concepts generated in 2004 are to feed forward into development and evaluation during 2005. These follow-on steps will be conducted under the aegis of the WIDE 6.3 project. As of the time of this writing, a first cut prototype illustrating timeline tool functionality is expected to be available before summer 2005, and a structured evaluation process is tentatively planned for the November 2005 timeframe.

After the January 2004 TIM, Gina Thomas-Meyers and Randy Whitaker met again to review and finalize the first edition of the timeline tool design concepts. We agreed that the concept specifications laid out in the December 2004 presentation to the TACC clients were generally sufficient to use as a starter set. We reviewed these concepts so as to identify any deficiencies or details that needed to be clarified. The following details were cited for clarification in this final review:

- In both the multi-mission and individual mission displays, the Mission ID shown in the header block will be the Mission ID for the first of possibly multiple sorties depicted for multi-leg missions.

- The departure and arrival times depicted in the timeline display (e.g., to either side of the graphic central display) will be the times associated with the sortie that is selected (in the multi-sortie case).

- If no sortie is currently selected:

  – If the current time indicator is on-screen, the times displayed will be the times associated with the sortie that intersects the current time indicator.

- If the sortie is on-ground relative to the current time indicator:

  – If there are one or more pending sorties within the given mission, the times displayed will be those for the next sortie (relative to current time).

  – If there are no pending sorties remaining within the given mission, the times displayed will be those for the last sortie completed.

- If the current time indicator is off-screen (e.g., the user is looking 2 days out into the future):

  – If the mission is ongoing in the timeframe displayed on-screen, the times displayed will be those associated with the first (leftmost) visible sortie.

  – If the mission has been completed, the times displayed will be those associated with the last sortie.

- The mission set selection menu in the multi-mission display needs to include a category for 'User Defined' sets.

- The User Defined mission set selection entry needs to make provision for subselection of one out of possibly multiple user-defined sets (as time goes on).

- In the Core display, the DIP period indicators displayed will be only those associated with the nations currently displayed on-screen. This is done to prevent confusions in addressing a DIP indicator associated with an off-screen overflight.

Additional Timeline Tool Design Features Not Included in the First Version

In the course of the timeline tool design work, some concepts were generated that were not included in the first edition of the design specifications. The first of these was a general display protocol intended to better tailor individual timeline tool display presentations to the immediate use situation. The second was a candidate approach to handling a type of temporal display that our TACC customers repeatedly cited as useful in our December 2004 interviews.

The first issue concerned how one might configure the individual timeline tool display to more effectively cue the user on pending alerts impinging on his / her decision space. The first edition design concept provides for alerts to be cued at 3 levels of referential granularity:

- At the level of individual lines within a cluster (color-coding of visual elements on a single line).
- At the level of each individual cluster (color-coding of the 'end tabs' on the left and right margins of the cluster element).
- At the level of the overall mission as summarized in the Core (color-coding of both visual elements within the Core visualization as well as the 'end tabs').

In the event a 'red alert' condition is flagged on a given line within a cluster (and hence on the overall cluster representation and the Core) the user needs to be able to efficiently locate the detailed data associated with the alert condition. Furthermore, he / she must be capable of evaluating it relative to other mission parameters. There are multiple clusters vertically 'stacked' in the individual display, and any combination of one or more clusters may be flagged in the cued alert condition. To evaluate the alert condition, the user needs to be able to readily locate the relevant cluster(s) (and line(s) therein) and correlate their implications with the data in other clusters and the Core.

We propose a strategy in which any cluster flagged with an alert is 'cloned' and its copy migrated to the top of the display (above the Core element). This means that under an alert condition the set of 'non-green' clusters will be replicated as a set above the main Core. The point of this manipulation is to provide a user with a ready summarization of any affected clusters at the point the individual timeline display is opened and at every point where there is a change of state thereafter. The set of 'cloned' clusters above the Core focus the user's attention on the specific topics relating to the alert condition, and they thus constitute a dynamic circumscription of his / her immediate scope of concern.

This tactic requires the ability to independently replicate and move cluster elements within the individual timeline palette. It also presumes an ability to track which clusters need to be cloned and (re-)displayed above the Core at any given time. Because we felt these capabilities could only be usefully explored once a basic timeline prototype was developed, we elected to defer inclusion of these features in the design specifications.

The second major extension to the original timeline tool design concept concerns a different type of temporal visualization than we originally considered. One of the recurrent themes in the feedback received at TACC in December 2004 was the utility of being able to track a specific resource or asset through time - regardless of its incorporation in one specific mission. The two most commonly cited examples were those of a 'tail number' (specific aircraft) and a specific piece of cargo.

The original timeline tool concept was configured for the visualization of multiple resources intersecting in the composite set of things necessary to conduct one given mission. What was being cited here implied three features which diverged from the original timeline tool design concepts:

- The primary element being plotted on the timeline would be a single asset

- The timeframe over which it was being plotted would exceed that of any single mission

- No single mission could serve as the referential basis for the display

The general form of a timeline display could certainly suffice for this variant type of visualization capability. However, the longer timeframe capability would make for a display in which horizontal scrolling might be more extensive and more cumbersome should the user need to trace the resource across its entirety. As a result, we sketched a variation on the timeline format in which the illustrated temporal span would have to be segmented and 'wrapped' (analogous to the manner in which text is wrapped in a word processor). This in turn implied we would have to configure the 'resource timeline' as a stacked set of subsidiary display elements, each one of which would represent the asset's history over a particular length of time. This 'stack' of constituent timeline elements would resemble a multi-mission display or the set of clusters subsumed within an individual display.

Although this variant - to which we gave the working label 'resource timeline' - could be readily envisioned, its form is sufficiently distinct to warrant additional evaluation and design work. As such, we have deferred further elaboration of this concept in favor of concentrating on the initial editions of the individual and multi-mission timeline tool displays.

Summary

This chapter has provided only a summary overview of the WCSS design support efforts conducted under the aegis of the WIDE 6.2 project. Proceeding from background knowledge acquisition to design concept presentation and feedback capture in the space of only 6 months made this portion of the WIDE 6.2 itinerary quite labor-intensive. By working in a stepwise manner with many team members participating, we were able to

generate both the basic design criteria and an extensive set of design concepts in a relatively short time. We believe the results have justified this investment of time and effort.

This work will feed directly into the WIDE 6.3 development and evaluation work scheduled to continue through FY05 and into FY06. Beginning with April 2005, the designer previously operating on the WIDE 6.2 contract (Dr. Whitaker) will migrate to the ongoing WIDE 6.3 project as a subcontractor. All other personnel will remain unchanged, and WIDE design team continuity will be assured.

Chapter IV.
Reflections on Work-Centered Support Systems
(WCSS Methodology Development)

Introduction

Work-Centered Support Systems (WCSS) has emerged in recent years as a framework or philosophy intended to guide the design of software systems. The framework is compatible with many other cognitive engineering approaches (e.g., Klein, *et al.,* 1997; Rasmussen, Pejtersen, & Goodstein, 1994; Woods & Christoffersen, 2002), but is intended to be more comprehensive, serving as a tool for members of the design team to communicate with each other and with those working outside of the design process. As WCSS has evolved, proponents of this approach have begun to articulate a design process termed *Work-Centered Design (WCD).*

This paper describes a project aimed at eliciting the experiences of those who have been involved in the development and implementation of the WCSS philosophy and associated design process across a range of projects. The primary goal was to reflect back what has been learned from those who have been immersed in WCSS ideas, so that the ideas and experiences from this line of research can be articulated and examined. Specific objectives include:

- Draw on the perspective of cognitive analysts new to WCSS to look at the philosophy and design process from the outside in
- Identify relevant techniques, methods, and artifacts associated with the WCSS philosophy
- Review and document successful WCD applications within the context of the WCSS design philosophy

WCSS/WCD Overview

In 2000, Eggleston and his colleagues identified three defining characteristics of WCSS. First, each WCSS must use both direct and indirect methods of aiding. The system should provide direct aiding by drawing the users' attention to pertinent situations or problems, and it should provide indirect aiding by presenting and organizing information in an easily accessible format. Second, each WCSS must provide tailored and context-sensitive support. In other words, the aid must provide support as needed depending on the current context and state of events. Third, there should be a single organizing framework. Although the system may incorporate a range of support elements, they must be integrated into a well-formed, single support system. In fact, it could be said that "the entire interface is treated as the aid" (Eggleston, 2003).

WCD emerged a few years later (Eggleston, 2003) as WCSS researchers began to reflect on their own practices and articulate a design process that would facilitate the development of the types of systems articulated by the WCSS philosophy.

The WCD framework focuses on supporting all elements of work including collaboration, workflow, decision making, and product development. Designers are encouraged to keep these elements in mind and even use them as a checklist throughout the project to insure that the resulting system supports these four key components of work. An overview of the WCD process is presented in Figure 19.

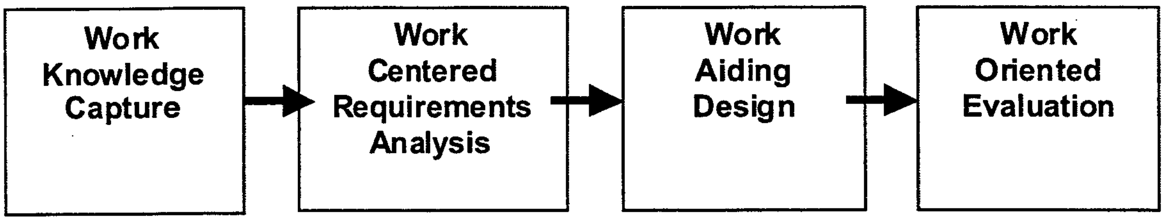


Figure 19: Overview of the Work-Centered Design (WCD) framework.

Method

The method used for this project combined several analytical approaches to reflect on the WCSS philosophy and WCD process. Literature pertaining directly to work-centered approaches was reviewed. These publications provide refined perspective on the work-centered approach to design. In addition, interviews were conducted with researchers to identify nuances of the approach that might not be captured in the formal publications. To pull the evolving nature of this approach into the analysis, exemplar projects were reviewed and compared using the WCD framework as the basis for comparison. Finally, investigators analyzed literature, interviews, and projects as the overall WCSS and WCD approach. Results reflect this holistic approach to the analysis.

Literature Review

A focused literature review was conducted. Researchers compiled papers, manuscripts, and briefing slides related to WCSS and WCD. All documents were examined for articulation of the WCSS philosophy, description of the WCD process, reference to specific instantiations of both, as well as allusions to relevant artifacts. Documents were compiled into two reference libraries. An electronic library was created for all materials gathered under this effort. A bibliography of all documentation reviewed is included at the end of this document. The second library was compiled as a means of culling exemplar artifacts. The artifacts in this library are organized according to the WCD framework, to illustrate representative documentation schemes used for each WCD stage.

Interviews

Individual interviews were conducted with four WCSS researchers, each of whom had been involved in at least one WCD project. Interviewees had a range of backgrounds, including two cognitive engineers, one user interface designer, and one software

designer. Each had been exposed to the entire WCD process and had participated to some extent in each of the stages describe in Figure 19.

Prior to each meeting, interview outlines were created, and participants were asked to provide any artifacts they created as part of the WCD design process. Each participant provided documentation for review. As appropriate, these documents were used during the interviews to assist in capturing information about the WCD process.

Interviews lasted between 1 and 2 hours and were conducted either in person or over the phone, depending on participant location. Each interviewee was asked to describe his/her role in each of the WCD projects in which s/he had experience. Interviewees were asked to share any intermediate artifacts that remained from the projects. In addition, a discussion of the unique aspects of each of these projects and of WCSS took place.

Exemplar Projects

Three exemplar projects were identified as efforts conducted within the WCSS philosophy. These included:

- Human Interaction with Software Agents (HISA) conducted from March to December 1999,
- Integrated Flight Management (IFM) conducted from June 2000 to April 2001, and
- Global Air Mobility Advanced Technology (GAMAT) conducted from February 2001 to September 2002.

Each project was examined via literature review and interview data for instantiations of the WCD process. Although the WCD process had not been articulated at the time the projects were conducted, discussions of WCSS and early writing on the topic were taking place. The projects were thus viewed as valuable examples which could be used to reflect on the strengths of WCSS as a guiding philosophy and from which observations about useful methods and artifacts could be made.

Data Analysis

Three collaborative analysis meetings were held in which investigators reviewed interview notes, information gleaned from literature review, and information gained from analyzing the exemplar projects. Analysis consisted of multiple sweeps through the data in which four categories of information were examined:

- elements that differentiate WCSS from other cognitive engineering approaches,
- artifacts from previous WCSS/WCD projects,
- strategies and methods described by WCSS researchers,
- and goals for future WCSS/WCD efforts.

Results

Examination of information collected on this effort focused on addressing the three project objectives of 1) examining the WCSS philosophy and WCD process from the outside in; 2) identification of relevant techniques, methods and artifacts, and 3) reviewing successful WCD applications. The results section is generally organized according to these three objectives. The first two subsections speak to findings associated with the *design process* and the third subsection addresses findings related to the WCSS *products*. Analyses found that there are characteristics or elements of the WCSS/WCD design process that are distinctive. Each of these elements, associated with the design process, is discussed. Additionally, evolution of the design process seeks to take advantage of artifacts created by the researchers. Therefore, artifacts are examined in some detail as aids in the design process. Finally, the purpose of the WCSS/WCD design process is to produce applications that support collaboration, work management, decision-making, and product development. Exemplar WCD applications are therefore examined.

Elements that Differentiate WCSS and WCD

Interviewees reported that the elements that differentiate WCSS/WCD from other cognitive engineering approaches primarily had to do with process. One highly visible element is that WCD addresses the entire software engineering process. As depicted in Figure 19, WCD describes the initial front-end information gathering needed to understand the domain and generate design recommendations, all the way through identification of requirements and generation of design concepts, to the evaluation of the resulting technology. This became an important talking point in many of the interviews.

The design process is often described in terms of a series of steps. Similar processes have been described by systems engineers, cognitive analysts, and process engineers. Generally a set of between 4 to 7 steps are articulated. Interestingly, many cognitive engineering approaches tend to focus on a subset of the process. This is not to say that cognitive engineers do not participate in all these steps or that all the steps do not occur in most cognitive engineering projects. Rather, the point is that many approaches focus their writing and discussion on a subset of the process. In fact, emphasis within this process lines up well with the tradition from which an individual approach has emerged.

For example, approaches that have grown out of the psychology tradition tend to emphasize and describe the knowledge elicitation (e.g., interview and observation techniques) and qualitative data analysis portions of the process. Approaches that have grown out of engineering paradigms tend to focus on representation issues such as how to map out important relationships in a large socio-technical system, as well as design elements that will lead to more efficient and error-free system performance. Practitioners whose work has been in the field of user-interface design tend to focus their writing and discussion both on the generation of innovative and usable design concepts, and how to

test and refine design concepts before they are implemented. Figure 20 illustrates these overlapping emphases from different traditions.

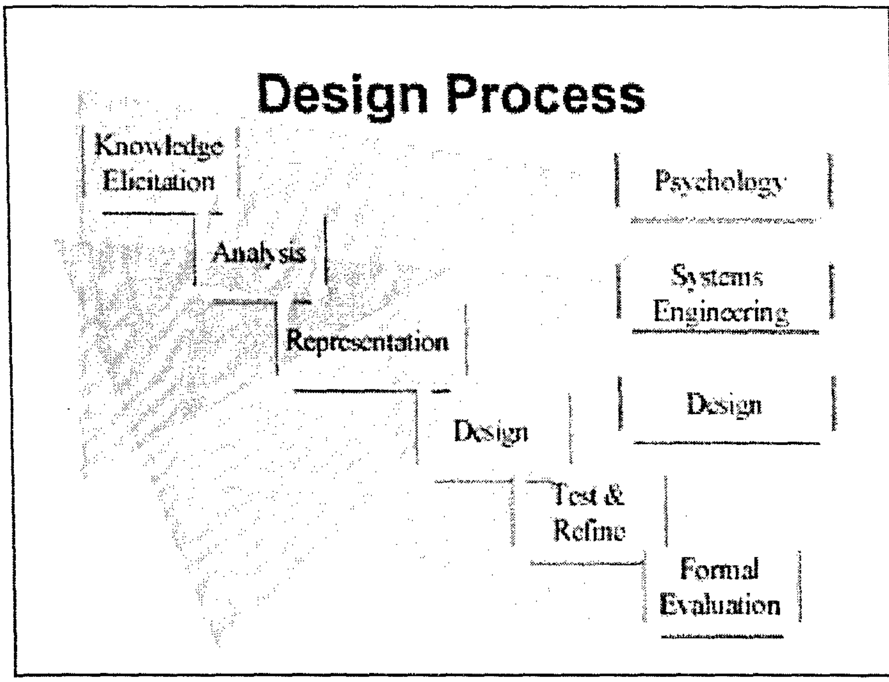


Figure 20:
Different Traditions Tend to Emphasize Different Portions of the Design Process

Continuous and iterative process

Practitioners working within WCSS have found the comprehensiveness of this approach to be valuable for a number of reasons. One important outcome of describing the entire process is that it facilitates a continuous and iterative process. When referring to processes such as the one depicted in Figure 20, practitioners generally emphasize that the steps are not discreet and that it is very common for the steps to overlap and even loop back to previous steps before moving forward. In spite of these assurances, however, the steps are too often treated as discreet. Often different companies are hired to accomplish different steps in the process, depending on their individual expertise. Communication among the teams addressing different steps in the process may be limited to shared documentation. It is not uncommon for team structure to preclude real collaboration between the different steps in the process.

Those involved in recent WCSS projects have had a different experience altogether. For these projects, the entire process has been articulated (as in Figure 19) and design teams have been made up of cross-functional elements and kept deliberately small. Each team has included members with different expertise, but all have worked together during each phase of the process. For example, a cognitive engineer generally served to lead and structure all the work-knowledge capture sessions. The plan generated for data collection, however included the user-interface designer, as well as the software designer.

By the same token, the user-interface designer took the lead in generating design concepts, but included team members with other areas of expertise in the process. The comprehensive nature of the WCSS process allowed each team member to anticipate stages of the design process they are not often directly involved with, increasing the likelihood that the team would both leverage information gained in previous steps and anticipate information and actions needed in next steps. This allowed for the kind of overlap and loop-back iterations commonly envisioned in the design process, but rarely practiced.

Work-centered design team roles

A second important outcome of describing the entire design process is that it encourages team members to better understand and appreciate the roles of other team members. For example, one interviewee reported that participating in knowledge elicitation sessions with a cognitive engineer helped him better understand the value of questions aimed at understanding workflow and workthreads throughout an organization. Further, the analysis meetings focusing on "leverage points" or aspects of work that might benefit from additional support were enormously beneficial to him in thinking forward to software design. In contrast, the cognitive engineer reported the value of having the perspective of the software designer early, during the work-knowledge capture, as the software designer was able to provide information about the effort and cost associated with proposed interventions (particularly as they related to obtaining the data needed to implement specific interventions). Articulation of the design process, combined with a small, cross-functional design team led to very effective collaboration across functional roles that is too rarely seen in design projects.

Communication among team members

A third important outcome of describing the entire design process has been effective communication between team members. Cross-functional teams often struggle to communicate effectively with each other as team members have different deliverables, different roles, and often different perspectives on the design process. Often information is "handed off" between different phases, increasing the likelihood that information will be lost or distorted. For example, it is not uncommon for cognitive engineers to be asked to perform knowledge capture and requirements analysis, and then deliver findings to a team of software designers. The WCD process has served to minimize these hand-off gaps by facilitating communication by all team members throughout the design process.

Effective design teams recognize the importance of communication in their interactions with one another. Efficiency and clarity of communication also plays an important role. This approach is evident in the WCSS philosophy as well. The WCD has been found to enhance communication among team members via several mechanisms.

One means of promoting communication is through inclusiveness. WCD teams have incorporated a small number of researchers who are involved in each phase of the design process, even if the phase is outside of the researcher's area of expertise. It was noted by

several team members that from the start, they were included in all phases of design, whether or not they were truly required for that phase. For example, the software designer was included in some of the Knowledge Capture data collection trips, although he was not the interviewer. This element of inclusiveness provides context to team members throughout the design process. Communication among the team members is facilitated by this contextual reference.

Another means of promoting communication is by encouraging researchers, within the Work-Centered Requirements Analysis phase, to analyze findings using methods familiar to their area of expertise. Team members were able to apply different skills and background, including European work analysis, participatory design, and theoretical mathematics to the various stages of design. The influence of background was most noticeable in the knowledge capture and analysis stages, where members relied on some of the more traditional representations associated with their background. For example, the user interface designer used the ethnographic approach of observation for work knowledge capture phase, and the cognitive analyst developed an abstraction hierarchy for the requirements analysis phase. This freedom to capture and analyze information in familiar terms allows for better and more efficient communication of design ideas to the team. That is, this allows the individual to collect his/her thoughts and perspectives, consolidate them, and communicate them to the design team concisely.

Yet another means of communication discussed by the design team is one of exploratory freedom. Team members discussed the fact that in the WCD process there was a period in the analysis and design phases whereby team members were free to explore design concepts with very little pressure to select one concept over another. This creative freedom allowed for the synthesis of ideas and further exploration of design solutions. The team also recognized that this period of freedom required specific checkpoints where the team had to select the better ideas for the design (e.g., sometimes these checkpoints were design review meetings). Milestone briefings often served as checkpoints and forced convergence at key points in the project. As with any creative work, much time was spent generating ideas, weighing priorities, and considering options. Milestone briefings forced the team to come together and agree on key issues needed to move the project forward. The exploratory period and the checkpoints both served as means of communicating user needs and design solutions among the team members.

The team also used artifacts as a means of communication. Artifacts seem to fall into two general categories. Artifacts that assist the individual team member in collecting and organizing their own thoughts and perceptions, and artifacts that assist in communication among team members. It should be noted that while the artifacts appear to hold a great deal of information regarding the information to be communicated, they do not seem to take the place of face-to-face type communication. They enhance the communication of ideas. In the WCD process, artifacts were generally considered to be a communication tool. In fact, when initially asked about artifacts the team reported good communication but few artifacts resulting from their work. Further review by investigators revealed that there was a rich set of artifacts associated with each WCD application. This disconnect

suggests that artifacts were used as tools used for design and communication, not necessarily products in and of themselves.

Work-centered evaluation

The fourth important outcome of describing the entire design process has been the opportunity to design and implement work-centered evaluation. Evaluation has been a challenge for the cognitive engineering community as few sponsors are willing to pay for an evaluation. For sponsoring agencies, user acceptance is often seen as the best indicator of success. While this is arguably a valid and pragmatic approach to the topic of evaluation, much can be learned from more in-depth evaluation strategies examining the broader impact of the new technology or system on larger organization. In addition to the difficulty involved in justifying a deliberate evaluation, measuring impact in terms of elements such as streamlined collaboration, enhanced workflow, higher-quality decision making, and improved product development is not a trivial issue. Well-established measures for these highly context-dependent elements do not exist. Baseline data for these elements are often not available or are very difficult to obtain. In fact, the question of what constitutes meaningful metrics for these elements has not been agreed upon.

In spite of these challenges, WCSS practitioners have used these projects to articulate a strategy for conducting work-centered evaluation. The GAMAT project in particular served as an exemplar for this approach to work-centered evaluation. In the context of this project, an evaluation was designed to examine usability, usefulness, and impact of the GAMAT system (Eggleston, Roth, & Scott, 2003). Further, the GAMAT evaluation was offered as an example of a comprehensive and cost-effective means to evaluate a prototype. In this case, usability, usefulness, and impact were assessed earlier in the design process than is typically seen. By assessing all of these elements during formative evaluations that occur iteratively throughout the design process, rather than waiting until a product has been fielded and a summative evaluation is planned, findings can be used to improve and refine the overall product before it is fielded.

Artifacts Leveraged in the Design Process

Artifacts are the output or products generated by the WCD design team. However, as stated earlier during the generation process artifacts serve primarily as tools for facilitating communication and compiling thoughts. WCD designers noted that the act of creating the artifact often has more value than the artifact product itself. The artifact was a communication tool for the team, but it did not contain all the information communicated during design discussions. It was also noted that many artifacts were most useful to the person who created it. Finally, the usefulness of artifacts was summed up by one interviewee in the following statement, "Everything is helpful in some cases, nothing is helpful in all cases."

Interviews with WCD researchers revealed that artifacts were central to communication of WCD ideas among team members and with customers. Given this thesis, it is reasonable to seek to identify artifacts that helped to facilitate communication among team members. These artifacts could then serve as sample tools for people new to implementing the WCD approach. Additionally, as these artifacts are created within a particular domain, they could be sampled or used as templates for reuse on similar programs. Note that careful reuse of artifacts would need to be employed to ensure that they truly apply to the work domain under study. In either case, identification of repeatable artifacts is an issue that continues to be considered by the developers of WCSS.

An interesting observation about the WCD process is that artifacts have not typically been prescribed for the designers. Each WCD designer brings or uses artifacts that are meaningful and helpful given their domain of expertise and their own experiences. While artifacts tend to fit into professional categories (e.g., knowledge elicitation, U-I design, software development), specific artifacts have not been prescribed for WCD design, they use what is useful for the project and meaningful to the WCD designer. They are *tools* used by the researchers to accomplish their goals and are therefore integral to accomplishing their tasks, but not necessarily seen as a bi-product of the task.

As will be shown in the sections below, the common frame of reference among team members and among design phases is the notion of work flows. Due to the integrated nature of specific work flows with WCD artifacts, many aspects of the artifacts are only totally repeatable within the context of the work. However, reuse of the framework or methods used to create the WCD artifacts is highly likely.

The sections below identify specific artifacts that have enabled WCD team members to compile their own thoughts as well as artifacts that were identified as having served as good communication tools among team members. Artifacts are organized according to the WCD phases of design.

Work Knowledge Capture

Artifacts resulting from the phase of the WCD approach seek to capture verbalizations of the people being interviewed, observations made by the design team, and processes employed by the organization under study. Resulting artifacts generally seek to organize this information in meaningful ways without changing or adding a great deal of analysis or interpretation. For example, Figure 21 identifies the first page of notes compiled after a typical knowledge capture trip. The notes begin with an overview of the objectives, followed by specific activities. The notes continue by identifying specific topics discussed during the knowledge acquisition interviews. These discussions are paraphrased in the notes with very few direct quotations from the users. In the end, some general areas are identified for further exploration. Interview guidelines are included at the end of the notes. A full example of these notes can be found in Roth & Scott (2003, March).

The communication benefit of knowledge capture artifacts used in WCD is to describe the data collection activity for future reference by the individual or the design team. Information gathered here represents information prior to any *in-depth* analysis of the user needs within the work domain.

Notes from KA Sessions with Flight Planners, Mission Planners, Dips Clearance Shop and Flight Managers
March 12 – 14, 2003
[illegible]

These notes summarize the results of interviews and observations that were conducted at the Air Mobility Command at Scott Air Force Base on March 12 – 14, 2003.

The objectives of the knowledge acquisition (KA) activities were to:

- Refine our understanding of Flight Planning process and opportunities to improve it
- Refine our understanding of the interaction between the flight planning process and the DIP Clearance process
- Identify how the flight planning process fits into the larger mission planning process
- Identify opportunities for a WCSS to support flight planning (encompasses dip clearance element)

The KA activities conducted during the site visit included:

Interviewing and observing the "Flight Planners" in the [illegible] ([illegible]). [illegible] was the primary user interviewed. The interview built upon a previous interview with him in December 2002. Additional discussions were held with Paul [illegible] and [illegible] provided us soft copy of the Permanent Bulletin Board (PBB) file consisting of routine flight plan ID's for "city pairs", and files from the "[illegible]" from [illegible].

Figure 21. Sample of Note Summaries
From Roth, E., & Scott, R. (2003, March).

Work-Centered Requirements Analysis

While the WCD philosophy identifies a Work-Knowledge Capture phase and the Work-Centered Requirements Analysis phase, it often difficult to distinguish between the two phases. Due to the iterative nature of these activities, they are often not separated in time. The main distinction of this work-centered requirements analysis phase, however, seems to be contained within the word *analysis*. Members of the design team clearly indicated that following the knowledge acquisition phase they integrated, compiled, sorted, and collapsed information in ways meaningful to their area of expertise. For example, the cognitive analyst looked at leverage points, the user-interface designer began conceiving design elements, the software designer began considering data sources and coding alternatives, and the evaluator began considering elements to test formatively and summatively. In the WCD approach each member begins to create artifacts appropriate for their area of expertise (e.g., software design). These artifacts are then used as communication tools for exploration. Freedom of communication enriches the analysis until a checkpoint is reached or the design phase is initiated.

The central organizing principle in many of the artifacts generated during this phase is the work flow. This common frame of reference allows the designers to work within their areas of expertise while maintaining a common foundation or set of artifacts to which they can collectively refer. Figure 22 shows a segment of a typical WCD work flow.

Overall Mission Planning Process

- Start with a rough cut at a flight – where it will stop, how long it will remain on the ground
- Put into CAMPS which generates an output that goes into GDSS – Form 59 for a mission
- The output of a mission plan is an 'itinerary' rather than a flight plan
- Since channel shop has regularly scheduled missions generally do not need to go to flight planners for 'what if'
- SAMS – will go into ACFP themselves
- Issues of concern to a mission planner include
  - MOG
  - Operating hours
  - Field closure
  - Fuel stops
  - Holidays that might result in field closures or embassy closures
  - Runway length
  - Fuel
  - There is a tradeoff between how much fuel you can carry in a plane and how much cargo, therefore refueling stops becomes a consideration
  - How heavy an airplane can be and still be able to take off from a particular airfield
  - Terrain
  - DIP clearance requirements -- Mission Planners attempt to be familiar with the Foreign Clearance Guide but the rules change with the change in politics [8]
  - Middle East DIPS can be hard to get but no easy way to go around it, because of the cost of gas and time to go around them [9]

Figure 22. Sample Workflow. From Roth, E., & Scott, R. (2003, March).

Many of the artifacts within each of the design phases leverage work threads or scenarios to describe or analyze information. The analysis phase is the primer for connections with the work flows. For example, Figure 23 shows a workstream with work-centered interface concepts, and links to related software systems. This figure encapsulates perspectives of cognitive requirements, related screens, and software links all using the work thread as a means of organizing the information.

Other examples of artifacts in recent WCD projects include abstraction hierarchies; lists of leverage points; work threads; major tasks; goal lists; lists of requirements (including elements of work and what makes it hard).

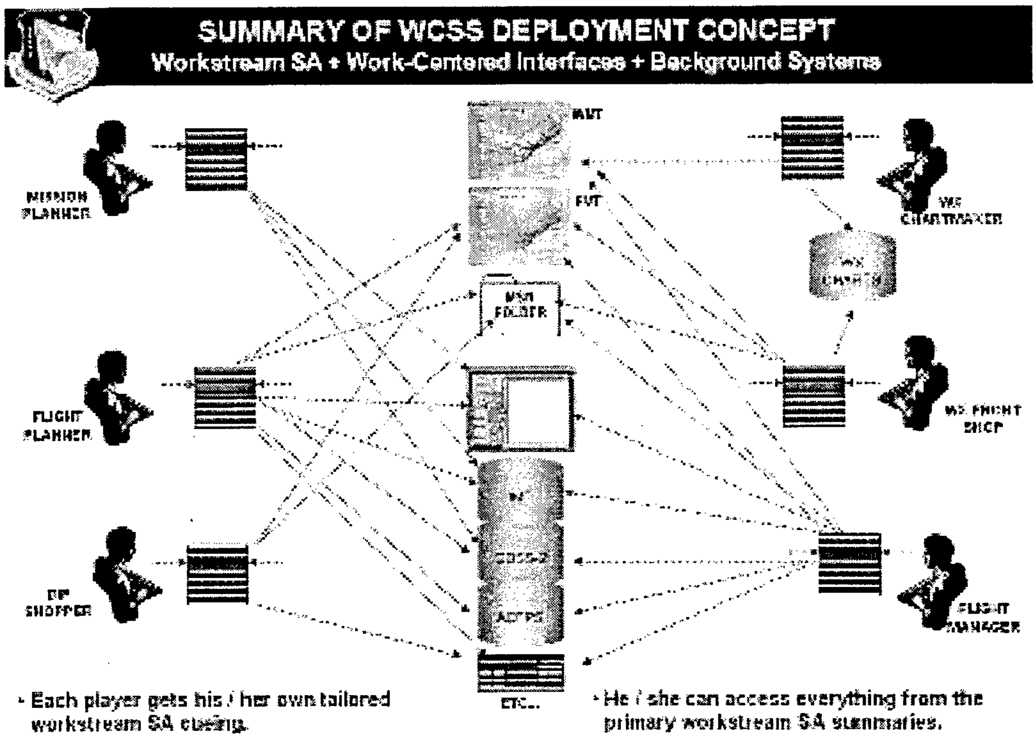


Figure 23. Workstream SA + Work-Centered Interfaces + Background Systems. From Whitaker, R. (2003, August)

Work-Aiding Design

Artifacts that are produced as a result of the Work-Aiding Design phase primarily consist of graphical depictions of screens or portions of screens to be presented to users (e.g., mock-ups and storyboards). What distinguishes the WCD method from others is incorporation of WCD design principles. The Work-centered ontology is a design principle emphasized in the design process. Basically this principle ensures that the user will not need to learn new terms associated with the tool, they will be provided with terms familiar to their work environment (Eggleston, Young, and Whitaker, 2000). While a formal ontology was not created in the exemplar projects explored in this report, efforts were made to leverage language and representations familiar to the users communities. Other principles such as the first-person perspective principle, the minimal set of referential contexts, and focus-periphery organization principle are also typically discussed by the designers and incorporated into WCD designs (Eggleston, & Whitaker 2002). Mock-ups and storyboards typically incorporate these design principles. For example Figure 24 provides an architecture for design, but includes many of the design principles mentioned (e.g., minimal set of referential contexts in the geographic view, and the focus-periphery organization principle).

As with all artifacts produced in the WCD method, the primary purpose is one of communication and all communication has the common foundation of being work centered whether it is a storyboard or a design concept it is presented within the context of the work to be accomplished.

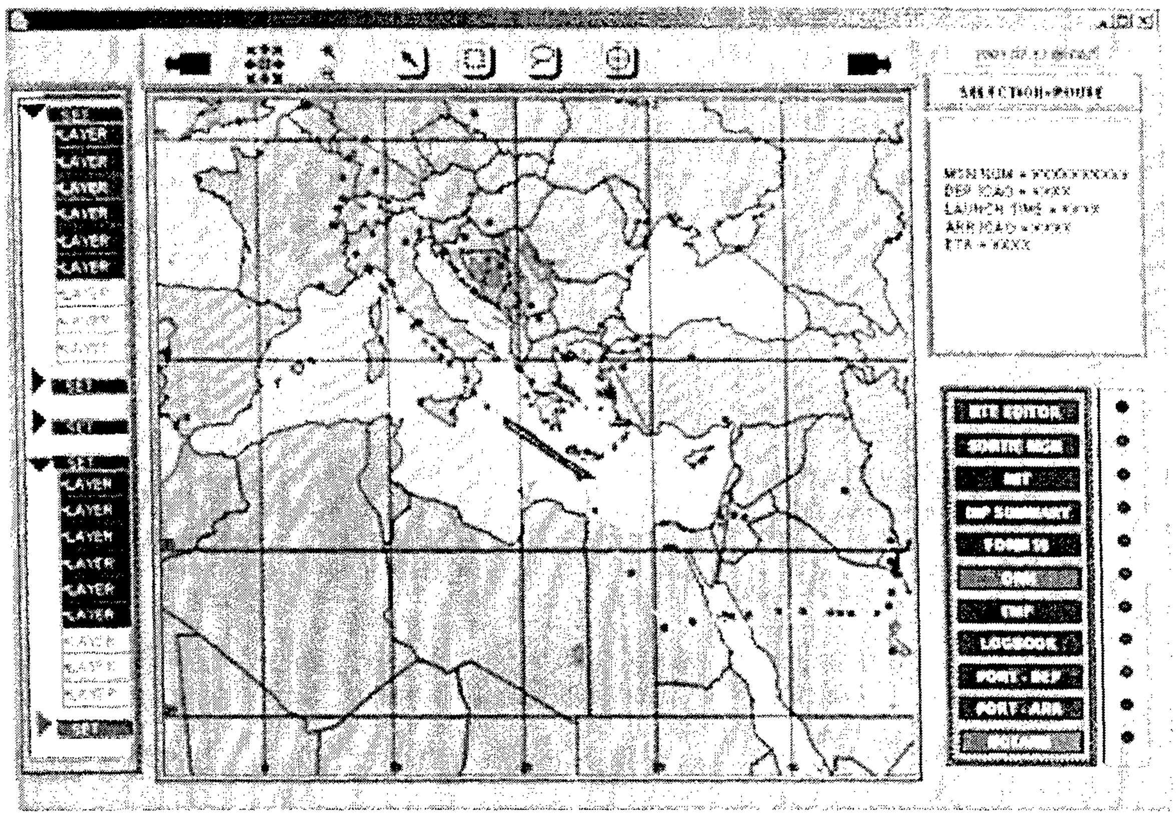


Figure 24. Basic Architecture for the GAMAT Flight Visualization Tool. From Kuper, S. (2004, February).

Work-Oriented Evaluation

Evaluation has been a challenging issue within the cognitive engineering community. Impact testing (e.g., value to the workplace) is generally very challenging with complex user interfaces, especially where situational awareness is a component of the interface. It is much more common to evaluate these systems with usability methods rather than methods that focus on usefulness or impact. The WCD process provides evaluative methods that collect information on all three of these factors. Artifacts, especially in the forms of their data collection tools, reflect this emphasis.

The framework for a WCD evaluation includes both formative and summative evaluations. There may be several formative evaluation opportunities throughout the design process and generally one summative evaluation. In any case, the method for collecting data includes elements targeted toward the three types of evaluation (usability, usefulness, and impact). There may be a warm-up period where the user performs certain

work-oriented tasks. During this period, usability information is collected. Following this sequence, another task may focus on usefulness. Finally post-test questionnaires and scales seek to identify impact of the software. Analysis from these sources focuses on triangulation of information, which is where information converges across data collection methods.

Again, as with the other phases, work-threads and samples of scenarios are critical in development of artifacts to support the formative and summative evaluation. The work threads and scenarios identified in the Work Knowledge Capture phase are vital to the effectiveness of the WCD evaluation results. Evaluations commonly select mini-work threads or sample scenarios to include as study tasks. The work-oriented connection back to the original requirements of the system and to each phase of the WCD process make the WCD evaluation process much more rigorous. The work flow allows all aspects of the design to interrelate. Figure 25 illustrates how the work flow is used in a work-oriented evaluation artifact.

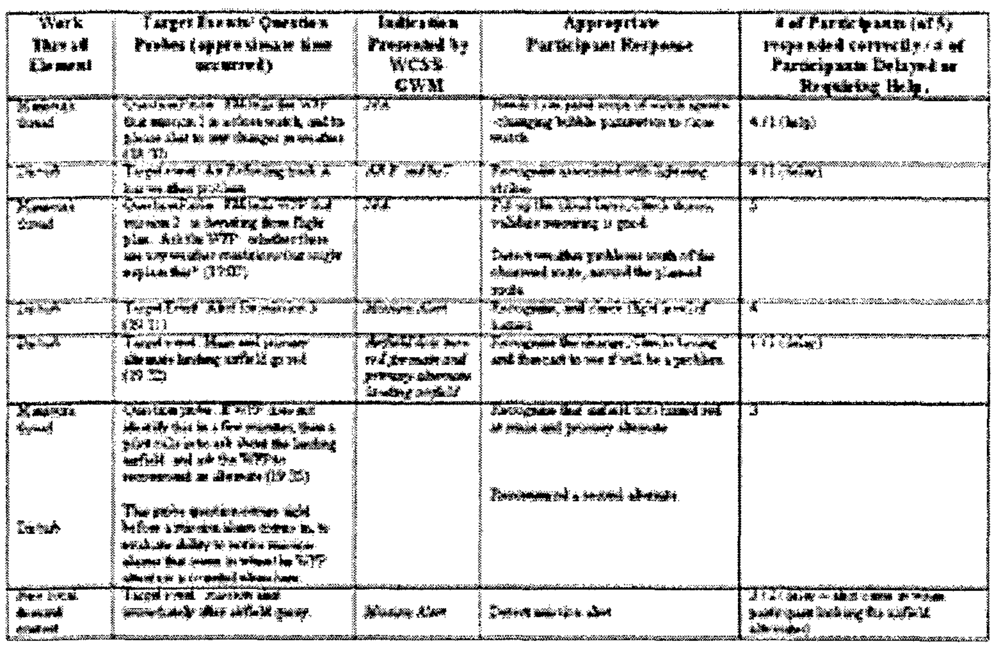


Figure 25. Extract From A Work Scenario Including Miniwork Threads Used to Evaluate Usefulness Of Prototype Aiding System for Weather Forecasters. From Eggleston, R.G., Roth, E.M., & Scott, R. (2003).

WCD Instantiated: A Review of Three WCSS Projects

In the following sections, three of the seminal WCSS projects conducted by AFRL (HISA, IFM, and GAMAT) will be introduced and reviewed. For each project, two types of descriptive exposition will be provided. The first will be a general overview of the project itself. The second will be a stepwise review of each project in terms of the four steps cited above for the WCD process path.

Human Interactions with Software Agents (HISA)

The HISA project (Mulvehill & Whitaker, 2000; Eggleston, Young, & Whitaker, 2000; Young, Eggleston, & Whitaker, 2000) began with a kickoff meeting in March of 1999. This project was the first opportunity WCSS researchers had to instantiate the design philosophy. In fact, the term WCSS had not been articulated when the project began, but the project served to help solidify and articulate ideas that became the basis for WCSS.

The project initially focused on developing software agents. This project was different from later WCSS exemplars in that there was not a stable, cross-functional team that worked together throughout the project. Two interviewees participated in this project: a cognitive engineer and a user interface designer. The team also included an additional user interface designer, a retired SME, and a series of software designers. Team members joined and left the project as needed. There was little support for continuity and collaboration throughout. For example, the user-interface designers were not included in observations and interviews during the front-end work-knowledge capture portion of the project. The software designers never had an opportunity to meet the user-interface designers face-to-face. Many of the obstacles to team coordination common to design projects were present for this team. It is interesting to note that in later WCSS projects, as the philosophy and WCD process were better articulated, many of these obstacles were minimized or avoided altogether.

In spite of many challenges within the design process, the HISA team developed a WCSS intended to support the development of Channel Plans at the Air Mobility Command's Tanker Airlift Control Center. Specifically, the WCSS focused on aiding users in dealing with issues associated with Maximum on Ground (MOG) restrictions. Intelligent agents were used to support a flexible interface that alerted users to changes in a number of conditions, as well as working MOG conflicts, need for Prior Permission Requests, and problem associated with ports (Mulvehill & Whitaker, 2000). The resulting system provided both direct aiding in the form of alerts, and indirect aiding in the design of the interface itself. The use of intelligent software agents allowed for tailored and context-sensitive support. All of the support elements were incorporated into a unifying interface.

In terms of process, this project may be the most difficult of the exemplars studied to map onto the WCD process. This is not surprising given that the WCSS philosophy was not articulated from the beginning and not shared by all the team members.

HISA Work-Centered Knowledge Capture

HISA's work knowledge capture was conducted primarily by two team members who then reported to the rest of the team what had been learned. The user-interface designer was not able to conduct knowledge elicitation first-hand due to limited access to Air Mobility Command. He relied on reports delivered via teleconference. These reports combined with a review of student manuals allowed the user-interface designer to begin to visualize the workflow. Reports from the two data collectors focused on error conditions, which was very helpful in determining how a WCSS might support and improve work processes. Additional data collection was conducted by the cognitive engineer and the lead software designer. These sessions focused primarily on workflow for the channel planner.

HISA Work-Centered Requirements Analysis

Data gathered during work-centered knowledge capture was represented as process flows for both channel planners and other types of single-use mission planners (i.e., SAAM, contingency). This information was used to generate a process flow for establishing a new channel map process, identifying portions of the process where software agents might be helpful. A communication interaction chart was also created to aid in examining collaboration across the mission planning process.

The user-interface designer relied on data gathered by others during the analysis phase. He was able to reflect on the data gathered regarding error states, combined with what he had learned about the work context of mission planners and different mission planning roles. It was the initial focus on error conditions that led him to frame the work to be supported in terms of three elements: port, passage, and package. This framework was maintained. As he moved into design work, these three elements became "an erector set for visualization."

HISA Work Aiding Design

During the design process, focus necessarily broadened from error states to the larger work context. The three elements of port, passage, and package held up as a useful framework for considering elements of work and information flow throughout the mission planning process. A draft sketch for a port planner was generated and shared with the team in June of 1999. This idea was refined and evolved into the Port Viewer concept (Figure 26), which served as the basis for subsequent demonstration prototypes. It is important to note that other interface concepts were proposed and considered during this time. A geographic display was proposed as a primary interface element, but later rejected. Although on the surface mission planning may seem to be an activity based on geography, the user-interface designer was able to explain to the team that setting up a

mission is a schematic and abstract process that would not be well-supported by a detailed, geographic display.

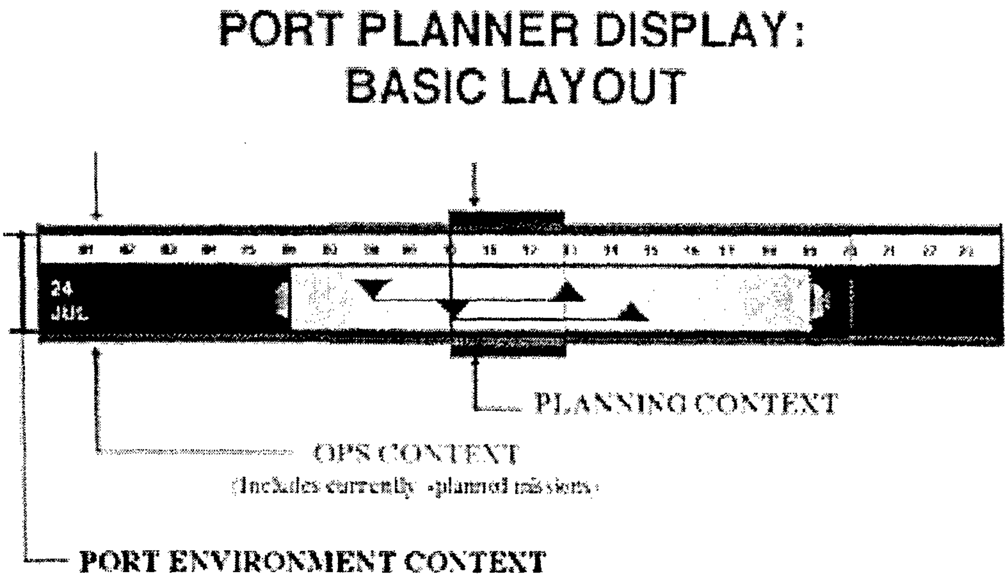


Figure 26: Port Planner Display

HISA Work-Oriented Evaluation

Our interviewees' involvement in the HISA project ended in January 2000. A set of detailed specifications had been generated, as well as storyboards illustrating the interface. In a briefing in January 2000, a commanding officer at the Air Mobility Command's Tanker Airlift Control Center was exposed to the Port Viewer concept. He expressed enthusiasm for the product and requested that it be built. Although there was no formal work-oriented evaluation of the design concepts, acceptance by the user community is a strong indicator of success.

Integrated Flight Management (IFM)

The IFM project took place in the context of a high profile transformation within Air Mobility Command (AMC). AMC was in the process of modernizing their approach to mission planning and flight. A reduction in the number of aircrews available was driving the need for a more efficient and less aircrew-intensive process. Two elements of this transformation were highly relevant to the IFM project. First, as part of this transformation, an Integrated Management Tool was introduced. The Integrated Management Tool was a software tool intended to support the transformation within AMC, and was touted as a success. The IFM team would be required to the use the Integrated Management Tool as a starting place and find ways to introduce work-centered elements into an existing software system. Second, job roles were changing somewhat dramatically. A Flight Manager position was added to reduce the amount of pre-flight work required of the aircrew and to provide addition support during flights. When the IFM project began no Flight Managers had been hired. As a result, there were no

experienced Flight Managers the IFM team could rely to explore workflow and the difficult aspects of the Flight Manager's job.

The design process in the context of IFM maps more closely to the recently articulated WCD. One interviewee described the IFM project as "a classic example of work-centered design process, if anything is."

IFM Work-Centered Knowledge Capture

A cognitive engineer and a user-interface designer worked together on this project to conduct knowledge acquisition. Observation sessions took place over the span of three days. They were able to observe all shifts, hand-offs between shifts, high workload periods, and typical workload periods. During these observations, investigators were asked to keep in mind that currently a busy day might require handling five flights. The projection was that each Fight Manager would handle 20 flights per day in the future. The cognitive engineer also had the opportunity to attend training sessions for the new Flight Managers.

A field observation report was generated to document what was learned during observation session. The team also had access to formal process flows created to describe the projected process after the introduction of Flight Managers.

IFM Work-Centered Requirements Analysis

Analysis revealed that the formal process flows of the projected process were of little value. The process flows depicted a single-incident in isolation. Without realistic context, the process flows provided a degraded view of the work. Further, no information about challenges or difficult elements of work, changing variables, information needs, or parallel processes was visible in the process flows.

The team found that they relied on post-observation "hot wash" sessions for preliminary analysis and design. This was an efficient way to review what had just been learned and discuss implications for the work process and potential design concepts. In a more formal analysis effort, the team looked for new ways of capturing process flow. They focused on rules, and under what circumstances the rules did not hold up. This analysis led to a better understanding of elements that create challenges within the flight management process.

IFM Work Aiding Design

The design concepts proposed for IFM focused on the new Flight Manager position and how to provide support that would integrate the Flight Manager into the AMC workflow. Recommendations included elements to support information sharing such as dual-layer logon privileges, as well as new software tools to support flight planning. These include the Flight Planning Guide (Figure 27) and the Planner Palette (Figure 28).

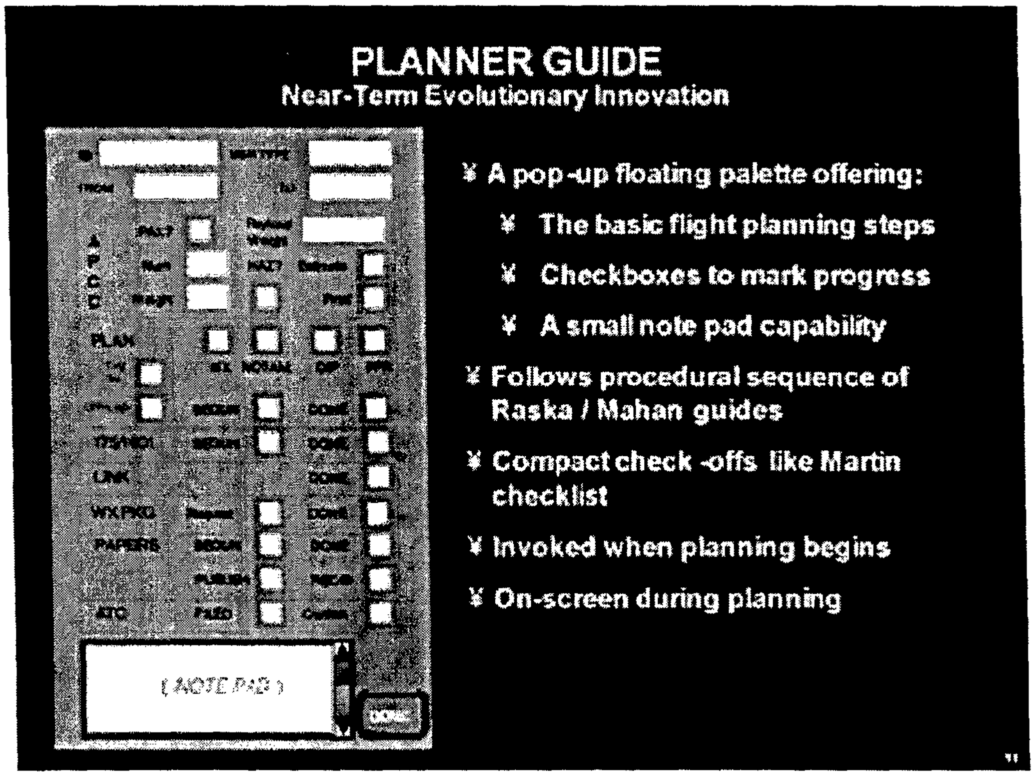


Figure 27: Planner Guide

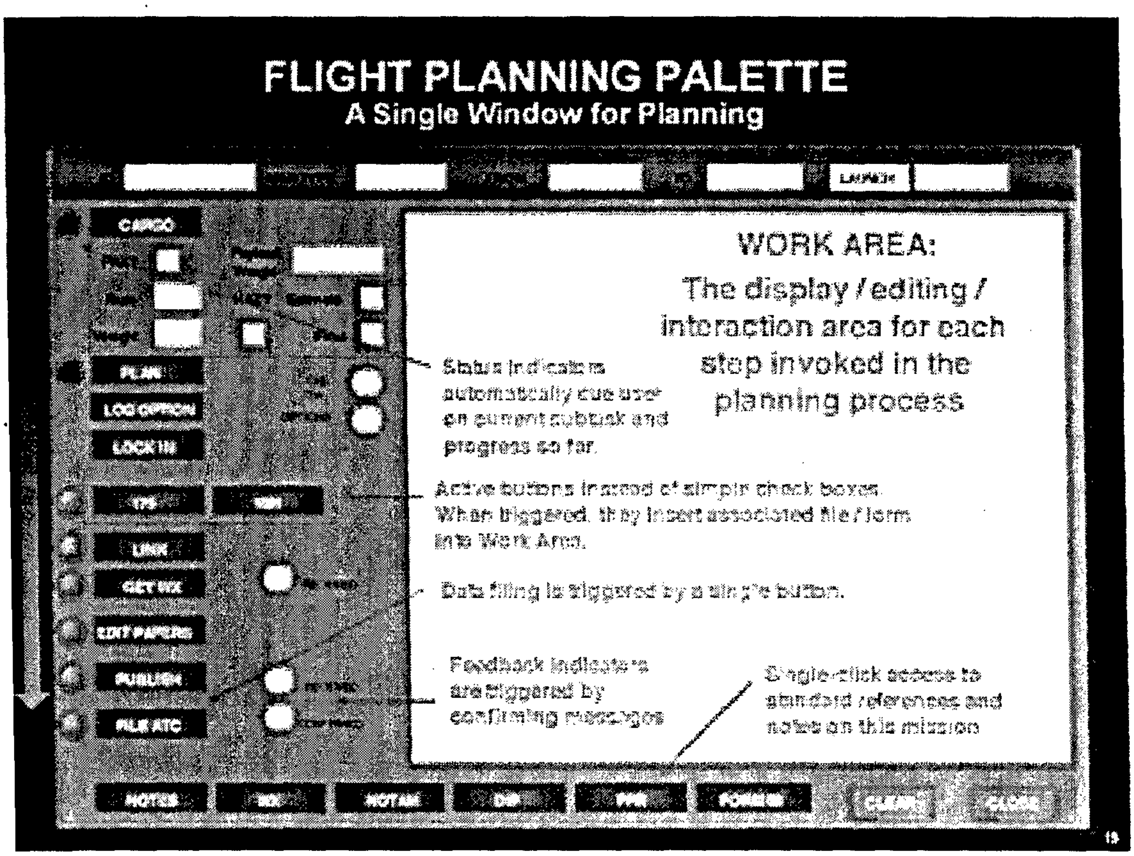


Figure 28: Flight Planning Palette

IFM Work-Oriented Evaluation

Our interviewees' involvement in this project ended with delivery of design concepts. No formal work-oriented evaluation was conducted.

Global Air Mobility Advanced Technology (GAMAT)

The GAMAT project (Scott *et al.*, 2005) began in Feb 2001, at which point the WCSS philosophy had been articulated and described in several papers (i.e., Eggleston, Young & Whitaker, 2000; Young, Eggleston, & Whitaker, 2000). This project differs from the others in that team was able to plan the GAMAT project with WCSS goals in mind, and as such offers perhaps the strongest example of a WCD process. The core team for this project was made up of two cognitive engineers, a user-interface designer, and a software designer. Within this team of four researchers, two had been fully involved in both HISA and IFM and were thus already entrenched in discussions of the WCSS philosophy, how to define it, how to bound it, how to differentiate from other approaches, and how to describe it in terms of a design process.

Similar to the other exemplars, the GAMAT project focused on software support tools for Air Mobility Command. This project directly addressed the weather forecasting and monitoring element in a military airlift service organization. The work to be supported in this case included pre-flight and enroute flight management as conducted by flight managers in collaboration with weather forecasters and pilots. Specifically, the focus of the effort was on "developing an intelligent system to aid near-term weather forecasting in support of planning and managing airlifts" (Scott et al, 2005).

In the GAMAT product, a map display is used as the framework for the interface and provides indirect aiding via a number of mechanisms (Figure 29). The user is able to adjust the display as needed with pan and zoom controls. The map can be tailored with different layers of flight and weather information, flight plans, satellite images, etc. The information on the map can be easily adjusted real-time as the situation and individual information needs dictate. The pending mission summary listing first proposed in the context of the IFM project evolved into the Sortie Palette during GAMAT (Figure 30), providing an overall summary of key information including missions of interest that can be sorted and organized as needed, and viewed in varying levels of detail. The Flight Planning Palette is integrated into the map display, presenting one cohesive interface.

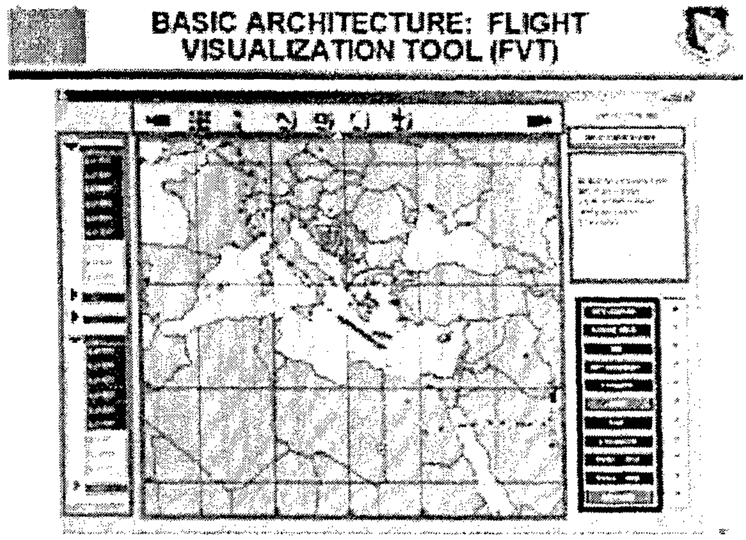


Figure 29.
A Map Display Served as the Basic Architecture for the GAMAT Flight Visualization Tool.

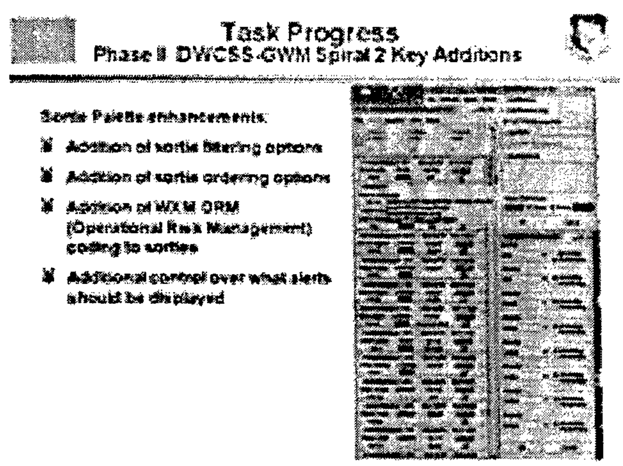


Figure 30: The Pending Mission Listing from the IFM Project Evolved into The Sortie Palette during the GAMAT Project.

Software agents provide direct support using intelligent automation to monitor missions and notify the forecaster when relevant changes occur. An important element of the GAMAT product is that the user can create, monitor, and modify these agents depending on which missions and geographic regions s/he would like to monitor.

GAMAT Work-Centered Knowledge Capture

A cognitive engineer led the work-centered knowledge capture of this project. A series of site visits provided observation and interview opportunities with flight managers and weather forecasters. All four core team members participated in some aspect of data collection. Prior to each data collection trip, an interview/observation plan was

generated, identifying interview topics and specific aspects of work to be explored. Interviews were conducted with personnel at several layers of management within the organization, as well as practitioners with differing levels of experience. Observations were conducted during different shifts and during both high workload and more typical workload situations. Later data collection trips included interviews that focused on user reactions to design concepts and storyboards.

GAMAT Work-Centered Requirements Analysis

Several interviewees reported that the work-centered knowledge capture and work-centered requirements analysis portions of this project merged, which is not surprising given the iterative nature of these two phases. While it may not be possible to distinguish two phases that were separated in time, it is possible to distinguish knowledge capture activities from analysis. For example, after each data collection trip, individual notes were typed and sent to the cognitive engineer who compiled notes into a central document. This document included a description of all data collection activities, key findings, implications for design, and open issues to be explored in future data collection. The document was then used as a frame for additional analysis activities that occurred during telephone conferences.

Other analysis activities included the generation of an abstraction hierarchy to better conceptualize the work domain and relationships. Depictions of work threads were generated to capture the flow of work throughout the organization (Eggleston & Roth, 2003). Lists of major tasks and associated goals were created. All of these analysis activities led to a list of work-centered requirements.

GAMAT Work Aiding Design

Analysis activities revealed a range of leverage points or opportunities to provide support. Design activities examined strategies for supporting decision making and product development, support for collaboration, integration of weather and flight information, and work management. A geo-referenced map was selected as the primary referential context. Other important dimensions such as time and mission were represented as an overlay to the map.

GAMAT Work-Oriented Evaluation

The GAMAT project provided the first real opportunity to conduct a work-oriented evaluation. A cognitive engineer led the evaluation effort, generating a test plan that included assessment of usability, usefulness, and impact. Work threads were used an important organizing feature of the evaluation. Realistic scenarios depicting a range of work threads were developed to provide important context for the evaluation (Eggleston & Roth, 2003).

Conclusions

Articulation of the WCSS Philosophy and WCD process has come a long way in a few short years. By leveraging prior work in the areas of cognitive engineering, human-centered computing, and user-interface design, WCSS researchers have successfully articulated a comprehensive approach to work-centered system design. Publications on the topic have described WCSS products as well as a WCD process intended to guide practitioners in achieving WCSS goals (Eggleston, 2003; Eggleston, Roth, & Scott, 2003; Eggleston, & Whitaker, 2002; Eggleston, Young, & Whitaker, 2000; Mulvehill, & Whitaker, 2000; Scott, Roth, Deutsch, *et al.*, 2005; Scott, Roth, Malchiodi, *et al.*, 2003; and Young & Eggleston, 2002).

The goal of the project detailed in this report was not to explain the WCSS philosophy and WCD approach, but to elicit the experiences of those who have involved in the development and implementation of the WCSS philosophy and WCD process. The goal was to reflect back what has been learned throughout the WCSS and WCD design processes from those who have been immersed in these ideas. Several clear benefits of the WCSS/WCD approach emerged.

Current Benefits

This section will identify and discuss the most commonly cited benefits of WCSS and the WCD process.

Comprehensiveness

Perhaps the most frequently discussed benefit was the comprehensiveness of the WCD approach. The comprehensive package encourages a continuous and iterative design process that is frequently envisioned but rarely occurs. The comprehensive package also allows a cross-functional team to exploit the expertise of each member throughout the design process. In the context of describing and developing WCD as a complete design process, a strategy for work-centered evaluation has been articulated. This strategy includes both formative evaluation to be conducted as part of iterative design, as well as summative evaluation to assess the usefulness, usability, and impact of a ready-to-be fielded product. This evaluation strategy takes advantage of information gathered during work-centered knowledge capture to develop realistic scenarios, tying together elements of the design process that are often separated in time. Throughout the complete process, the WCD framework promotes communication among and between team members.

Leveraging Traditions

Another benefit of the WCSS philosophy and WCD process is that it encourages use of a range of methods, artifacts, and expertise from a variety of traditions. Much like the

concepts of concurrent engineering or integrated product teams, many of the WCD applications have specified leaders for each phase of the design. These leaders are collaborating together during each phase of the process. Each leadership role reflects the area of expertise associated with that phase (e.g., the cognitive engineer leads the work-knowledge capture phase). Through incorporation of a variety of expertise, an in-depth understanding is attained in each phase. Additionally, the WCSS philosophy does not prescribe any particular methods or techniques within any phase of design. Therefore, the designer is free to use any methods or techniques needed for them to gain understanding of the designs required. For example, in several data collection trips, the cognitive engineer used several established cognitive methods for collecting information, while the user-interface designer used a different set of methods more suitable to user-interface design needs. Subsequently each individual produced artifacts that were unique to their own traditions and experiences. This element of acceptance in WCSS philosophy benefits the project goals by allowing designers to work with familiar tools and techniques to attain the goal of integrating ideas into a work-centered design for the customer.

Work-centered Checklist

Another benefit of WCD interviewees described was the ability to use specific elements of work-aiding as a checklist. Specifically, the WCD process includes consideration of collaboration, workflow management, decision making, and product development. Interviewees reported that they found articulation of these elements helpful throughout the design process. WCD practitioners were able to return to this list periodically to make sure all four elements were examined in the work context, represented in the requirements analysis, supported in the design, and taken into account in the product evaluation.

User focus Throughout

Interviewees indicated that the process assures user needs are being met. Through careful knowledge capture and definition of work flows, all subsequent activities can be traced back to the needs as defined by the end-user. For example, it is clear that many of the screen designs link to specific user needs as defined by the work threads and their related storyboards. Similarly, the work-centered evaluation relates specifically to mini-work threads to ensure that product is evaluated in the context of realistic challenges the user will likely face.

Facilitation of Communication

The WCD process provides a framework which creates bridges between the design phases. All too often design teams are made up of distributed players with different contractual and proprietary interests. It is not uncommon for user-interface designers to pass written specifications on to software developers, with little opportunity to collaborate in a meaningful way about the user context and the intent behind the specifications. The WCD process, in contrast, encourages the use of a single-cross

functional team that is involved throughout the process. This facilitates communication from one phase to another and among team members with different areas of expertise.

Future Benefits

In addition to the benefits of WCSS/WCD that are available today, interviewees identified several areas they would like to see developed in the WCSS of the future. These future benefits included:

- Further articulation of the principles of good design. While there are already several important principles generated by the WCD designers, others principles are likely to emerge as the philosophy matures.

- The notion of evolvable systems. This notion would allow users to change the design of their platforms while adhering to the design philosophies associated with WCSS.

- Strengthening of scientific foundations associated with design and development. Areas to be investigated further include theoretical, testable, inductive, and repeatable foundations of science.

    - Theoretical foundations include notions such as statistical versus analytical generalization.
    - Testable foundations include notions such as triangulation of results.
    - Inductive foundations are likely to be found in looking for similarities among work domains.
    - Repeatable foundations include the importance of replication or showing the same result across multiple evaluations (Eggleston & Roth, in press).

  In each of these cases, the WCSS philosophy already has evidence supporting these notions; further investigations will assist in determining the philosophy's ability to enrich the science in these areas.

Discussion

This exercise in reflection has provided interesting insights into WCD. Much of the writing about WCSS and WCD has focused on distinctive elements of this approach including:

- An articulation of what constitutes a WCSS product (Scott et al, 2005; Eggleston, Young & Whitaker, 2000)
- Reports of previous WCSS projects (Mulvehill & Whitaker, 2000:Goan, 2002)
- A high level description of the WCD process (Eggleston, 2003)
- Identification of work-centered design principles (Eggleston & Whitaker, 2002)

- A description of what constitutes a work-centered evaluation (Eggleston & Roth, in prep; Eggleston, Roth, & Scott, 2003)

Interviews and review of artifacts, however, uncovered elements of WCSS/WCD that have not yet been published. For example, work within the WCSS philosophy has raised issues about the value of repeatable artifacts. This is an issue the team continues to regard as an important one. Interviewees acknowledge the need for artifacts to guide the WCD process and to facilitate a repeatable process. As experienced designers, however, our interviewees recognize that few (if any) artifacts are applicable to every design problem. One element WCSS practitioners value is the freedom to employ methods and artifacts as they seem appropriate, rather than based on a prescribed process or dogmatic paradigm.

Further, several of benefits of WCD emerged as a result of interviews rather than literature review. The strength of this comprehensive approach in facilitating communication, providing links between design phases, and promoting smooth cross-function team performance was highly valued by interviewees – and no doubt contributed considerably to the success of recent WCSS projects.

It is not surprising that many of these important process issues arose via discussion rather than publication. Most publication outlets value project descriptions, products, and methods over detailed process or "how to" issues. Nevertheless, for an evolving paradigm such as WCSS, this sort of reflection and examination of how a WCSS is generated, how a team approaches each of the stages of the WCD process, and links between the two is key to progress. The WCSS movement has made great strides in recent years, leveraging strengths of multiple traditions to articulate a comprehensive approach to software design. Persistent instantiation of the principles articulated thus far, combined with continued reflection, will allow WCSS practitioners to address additional design challenges such as:

- How will the WCD process handle projects dealing with much larger systems, such as a future generation battleship? The team composition so far has been small with all roles covered by one expert in each area. Communication and artifact generation will be much more challenging in these large-scale environments.

- How will the WCD process handle closing the gap from early phases of R&D type design and development to full-scale development and implementation? In these cases, the team composition may change, the artifacts may be more stringent, and consequently, communication may be more challenging.

In each of these instances, the WCSS philosophy and WCD process will be stretched and perhaps strengthened as new challenges to the design process are encountered.

Bibliography: Materials Collected to Support this Study

Articles

Air Mobility Command. (1995). *Contingency and wartime air mobility management* (AMC Instruction 10-202, Vol. 3). Retrieved January 11, 2005, from http://afpubs.hq.af.mil

Air Mobility Command. (2000). *Tanker/Airlift Operations* (AMC Instruction 11-208). http://afpubs.hq.af.mil

Air Mobility Command. (2003). *Command and control (C2) responsibilities and procedures* (AMC Instruction 10-202, Vol. 2). Retrieved January 11, 2005, from http://www.e-publishing.af.mil

Air Mobility Command. (2003). *GWM Operations.* Cambridge, MA: BBNT Solutions LLC.

Air Mobility Command. (2003). *GWM Users' Manual.* Cambridge, MA: BBNT Solutions LLC.

Air Mobility Command. (2004). *Mission reliability reporting system (MRRS)* (AMC Instruction 10-202, Vol. 6). Retrieved January 11, 2005, from http://www.e-publishing.af.mil

Bisantz, A.M., Roth, E., Brickman, B., Gosbee, L.L., Hettinger, L. & McKinney, J. (2003). Integrating cognitive analyses in a large scale system design process. *International Journal of Human Computer Studies, 58*, 177-206.

Department of the Air Force. (2002). *TACC operations center handbook.* Scott Air Force Base, Ill.: Headquarters Air Mobility Command.

Department of the Air Force. (2002). *TACC operations center process flows.* Scott Air Force Base, Ill.: Headquarters Air Mobility Command.

Eggleston, R.G. (2002). Cognitive systems engineering at 20-something: Where do we stand? In M.D. McNeese & M.A. Vidulich (Eds.), *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa* (pp. 15-78). Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center (HSIAC) Press.

Eggleston, R.G. (2003). Work-centered design: A cognitive engineering approach to system design. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, Denver, CO, October, 2003*, 263-267.

Eggleston, R.G., & Roth, E.M. (in press). *Toward A Framework for Work-Centered System Evaluation.*

Eggleston, R.G., Roth, E.M. & Scott, R. (2003). A framework for work-centered product evaluation. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, Denver, CO, Oct 13-18, 2003*, 503-507.

Eggleston, R.G. & Whitaker, R.D. (2002). Work-centered support system design: Using organizing frames to reduce work complexity. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting, Baltimore, ME, Sep 30-Oct 4, 2002*, 265-269.

Eggleston, R.G., Young, M.J.& Whitaker, R.D. (2000). Work-centered support system technology: A new interface client technology for the battlespace infosphere. *Proceedings of the NAECON, 2000, Dayton, OH, 10-12 October, 2000*, 499-506.

Goan, T. (2002). *Facilitating work-centered interface development through patterns* (AFRL-HE-WP-TR-2002-0166). Wright-Patterson AFB, OH: Human Effectiveness Directorate.

Mulvehill, A. & Whitaker, R. (2000). *Human interaction with software agents (HISA)* (AFRL-HE-WP-TR-2002-0007). Wright-Patterson AFB, OH: Human Effectiveness Directorate.

Scott, R., Roth, E., Deutsch, S., Malchiodi, E., Kazmierczak, T., Eggleston, R., Kuper, S. & Whitaker, R. (in press). Work-centered support systems: A human-centered approach to intelligent system design. *IEEE Intelligent Systems.*

Scott, R., Roth, E., Malchiodi, E., Deutsch, S.E., Kazmierczak, T., Eggleston, R.G., Kuper, S. & Whitaker, R.D. (2003). Using Software Agents in Designing Work-Centered Support Systems. *IEEE Intelligent Systems (Special Issue).*

Sierhuis, M.& Clancey, W.J. (2002, September/October). Modeling and simulating work practice: A method for work systems design. *IEEE Intelligent Systems, 17*, 32-41.

Young, M.J. & Eggleston, R.G. (2002). Work-Centered Decision Support. *Proceedings of the NATO/RTO symposium on Information Management in Command and Control, Warsaw, Poland, 07-11 Oct, 2002*, 15-1 – 15-2.

Zhang, J., Patel, V.L., Johnson, K.A., Smith, J.W. & Malin, J. (2002, September/October). Designing human-centered distributed information systems. *IEEE Intelligent Systems, 17*, 42-47.

Notes

AFRL/IF. (2003, July). [AOC 06 POM Input]. Unpublished raw data.

Crawford, Lt. B. (2003, November). [K.A. visit to Scott AFB, AFRL/HECS]. Unpublished raw data.

Eggleston, B. (1999, May). [HISA combined knowledge acquisition document for knowledge acquisition sessions at the TACC]. Unpublished raw data.

Kazmierczak, T. (1999, September). [TRS DO-3 Meeting]. Unpublished raw data.

Kuper, S. & Kazmierczak, T. (2001, March). [Kickoff briefing and knowledge acquisition interviews for the GAMAT project, Work-Centered Support Technology]. Unpublished raw data.

Ragsdale, R. (1999, March). [Air mobility and tanker/airlift control center operations 101]. Unpublished raw data.

Roth, E. (2001, May). [CTA notes from observations and interviews of FM and WX, Work-Centered Support Technology]. Unpublished raw data.

Roth, E. (2002, February). [CTA notes from observations and interviews of WX, Work-Centered Support Technology]. Unpublished raw data.

Roth, E. (2002, December). [Flight planner CTA notes, Work-Centered Support Technology]. Unpublished raw data.

Roth, E. (2005). [Lists of planning factors and constraints]. Unpublished raw data.

Roth, E. (2001, July). [User feedback on GAMAT prototype presentation, Work-Centered Support Technology]. Unpublished raw data.

Roth, E., & Scott, R. (2003, March). [Notes from KA sessions with flight planners, mission planners, dips clearance shop and flight managers, Work-Centered Support Technology Development]. Unpublished raw data.

Roth, E., Scott, R., Kazmierczak, T. & Ragsdale, R. (2003, June). [Notes from KA sessions with flight planners, mission planners, dips clearance shop and flight managers, Work-Centered Support Technology]. Unpublished raw data.

Roth, E., Scott, R., Thomas-Meyers, G., Whitaker, R., Ragsdale, R. & Kazmierczak, T. (2003, September). [Notes from observation/interview sessions in support of WCSS for flight planning, Work-Centered Support Technology Development]. Unpublished raw data.

Roth, E., Scott, R., Whitaker, R. & Kazmierczak, T. (2003, November). [Notes from observation/interview sessions in support of WCSS for flight planning, Work-Centered Support Technology Development]. Unpublished raw data.

Thomas-Meyers, G. (2003, November). [Notes from Scott AFB]. Unpublished raw data.

Whitaker, R. (2000, December). [Field observation report: AMC/TACC flight managers]. Unpublished raw data.

Presentations

Britt, J. (2004, March). *Command & control for "Dummies".* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2003, October). *WCSS design methods: Challenges & thoughts.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2003, May). *WCSS technology: An update.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2003, August). *A work-centered approach to user interface design.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2001, May). *Human factors assessment of flight management work.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2001, April). *Human factors assessment of flight management work: Project review.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2002, April). *Work centered support system: Concept familiarization.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Eggleston, B. (2002, June). *Work centered support system technology.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

HISA Development Team. (1999, October) *Agent based decision support for mobility ops.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Kuper, S. (2004, February). *Global air mobility advanced technologies (GAMAT) ATD final program review.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Kuper, S. (2004, March). *Work-centered user interface technology and application for mission monitoring and hazard avoidance.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Kuper, S., Eggleston, R. & Scott, R. (2004, October). Using work-centered support system technology to enhance command and control. Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Militello, L. & Quill, L. (2005, January). *Work-centered support technology development: Project review.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Roth, E. (2003, November). *Work-centered support systems: A framework for design and evaluation of computer support systems.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Stanard, T. (2004, January). *User Interface Design Patterns.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Thomas-Meyers, G. (2003, December). *Using work-centered support system technology to enhance command and control.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Thomas-Meyers, G., Nagy, A., Kuper, S. & Wampler, J. (2004, September). *Choosing colors for work-centered support systems for command and control using a visual search task.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Whitaker, R. (2005, January). *Advanced information composition: WIDE 6.2 project.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Whitaker, R. (2003, August) *GAMAT work-centered support system (WCSS) review: AMC/TACC ("The Big Picture").* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Whitaker, R. (2005, January). *User interface design patterns: WIDE 6.2 project.* Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

CHAPTER V.
User Interface Design Patterns

Background

The proliferation of information and information technologies have transformed the nature of all large-scale enterprises. This includes those military enterprises whose functions constitute command and control (C2) operations. Typical categories of C2 tasks include operations planning, resource allocation, scheduling, coordination with external parties and units, executing operations, and monitoring the course of those operations. All these tasks have become more and more reliant on networked information technologies to process and present information employed by any given party or unit as well as to mediate communications among participants distributed in space and / or time.

AFRL's WCSS projects with Air Mobility Command (AMC) have been continuously conducted since the HISA (Human Interaction with Software Agents) project during 1999 - 2000. In the course of these projects our teams have sought to understand and analyze command and control operations in AMC's Tanker Airlift Command Center (TACC) and then to devise innovative user interface (UI) applications facilitating more effectiveness and efficiency in those operations. The two key challenges in our user interface (UI) design work to date have been:

- Obtaining a coherent and reliable body of knowledge about command center operator requirements, and

- Generating information displays and aiding capacities specifically tailored to allowing operators to meet these requirements with optimal task performance.

Achieving these twin objectives with respect to information systems is significantly different from the process of trying to do so with mechanical systems (e.g., the production tools used in manufacturing). Based on initial attention to physical or instrumental tasks, understanding operator requirements has been historically based on an analysis of the behavioral tasks a worker performs. The design and development of new tools or artifacts has accordingly been based on providing those instrumental capacities associated with the observed tasks upon which this understanding was based.

When dealing with information technologies rather than physical / instrumental technologies, behavioral task analysis is at best a limited means for understanding the intrinsic aspects of the work to be supported. Information-intensive tasks such as those we focus upon in our C2 projects are cognitive in nature. These tasks are often a matter of data interpretation, mental evaluations, and decision making. At the extreme, the only portions of the work process externally observable (and hence amenable to behaviorally-focused analysis) are the worker's interactions with the information system itself and the particular manipulations he / she undertakes in the course of employing the information

system to achieve his / her work objectives. Though these are certainly important aspects of the work activity, they fall far short of capturing the full range of task actions the subject is actually performing.

Much of the early work in UI design involved an attention to physical / instrumental factors inherited from prior knowledge in designing physical production systems. Conventional user interface designs therefore derive from application of the designer's training and experience derived from industrial practices and adherence to usability design guidelines. Such guidelines themselves are typically a hodgepodge of tips and 'best practices' accumulated from developmental experience, a set of prescriptions established by software vendors (e.g., Microsoft) to enforce compatibility with their existing products, or a combination of both. In other words, UI design is often conducted on the limited basis of 'what we've done before' or 'what we can do within the recommended protocols of a certain software environment'.

We need a better way to delineate and describe the work performed within C2 along with better guidance on how specific features of information technologies can support this work. The portion of the WIDE 6.2 effort reported in this chapter was dedicated to an exploration and evaluation of 'design patterns' and their potential for aiding us in making these complex design decisions more tractable.

In the course of this chapter we shall:

- Introduce the concept of a 'design pattern'
- Critically examine the applicability of 'design patterns' to information technologies in general
- Further examine the applicability of 'design patterns' to our work-centered design (WCD) methods and experiences
- Further examine the applicability of 'design patterns' to the work-centered support systems (WCSS) we design and develop
- Describe progress on an approach for specifying user interface design patterns related to the sort of C2 work functions we address in our TACC projects

An Introduction to Design Patterns

The concept of a 'design pattern' arose in the field of architecture - more specifically in the theoretical and applied design work of architect Christopher (Chris) Alexander. The origins of his attention to pattern and form in design date back to his work in the early 1960's (Alexander, 1964). Over the course of more than a decade, Alexander's explorations of form and regularity in successful and aesthetically-pleasing architectural designs led him to note the recurrence of certain 'patterns' descriptive of the manner in

which particular design features were consistently applied (Alexander, 1979). For example, the feature of a 'door' recurs in varying but essentially stable form across all instances of walled buildings and similar enclosures. Though there are many types of doors (and doorways, gates, portals, etc.), they all serve to connect the interior and exterior spaces in a manner which is at once physically tractable for implementation, readily usable, and consistent with relevant features of the usage context (e.g., cultural norms and symbolic attributes).

The accumulated set of such patterns describing common elements of a particular deployment or design space constitute a 'pattern language' (Alexander *et al.*, 1977). There may be distinct pattern languages applicable to each of any number of such spaces. Based on this, there is often reference to the 'pattern language' for a given class of application domain, a particular environment of deployment, and / or a specific product or artifact. The construct of 'pattern language' connotes a coherent syntax or feature set which is broadly descriptive of the most general features represented (or representable) in whichever of these contexts it is invoked.

Alexander's concept of design pattern has proliferated widely outside its parent field of architecture. Wherever designs provide a specific functional solution to a particular problem or situation, practitioners have found it useful to consider their products in terms of design patterns. With respect to information technology specifically, there was a surge of interest in design patterns during the 1980's in addressing commonalities among window-based user interfaces. Another surge of interest came in the mid-1990's with regard to the design of websites. Even more recently, software programmers have begun to describe code modules (e.g., routines, applets) in terms of design patterns. In fact, software architects generally prefer to work with proven pattern-based solutions, capitalizing on reuse of well-developed code. This minimizes the quantity and severity of errors, maximized productivity, and encourages the development team to concentrate on the task at hand -- development of software and system solutions, instead of debugging novel implementations for standard problem types.

It is interesting to note that Alexander's primary works on the notion of design patterns arrived in the late 1970's - just in time for the explosion of interest in information technology (IT). It is no surprise, then, that as time went on IT researchers began to examine Alexander's design patterns as possible conceptual bases for wrestling with the complexities of their own burgeoning field. With regard to our own work, there have been attempts to delineate pattern languages for user interfaces and supporting technologies. In the next section, we shall examine the most relevant such efforts.

What is a Design Pattern?

In one form or another, a design pattern is typically defined to be a descriptive specification for a recurrent solution to a problem. This may seem general and abstract, and these qualities are both evident and intended in the design pattern literature. For example, Tidwell (2000) circumscribes patterns' generality by claiming they, "... are not abstract principles that require you to rediscover how to apply them successfully, nor are they overly specific to one particular situation or culture. Instead, they are somewhere in between: a pattern describes possible good solutions to a common design problem within a certain context, by describing the invariant qualities of all those solutions."

In the original sense in which Alexander outlined them, patterns are framed at a level of generality sufficient to encompass a range of specific solutions in terms that allow them to be applied broadly. On the one hand, this generality is one of the attributes which have made design patterns so attractive in so many different fields. The concepts and constructs are sufficiently abstract as to be capable of projection onto most any object of design or deliberate innovation. On the other hand, this generality has allowed writers within these various fields to modify and / or extend their working definitions of a design pattern to the point that no two necessarily map onto each other.

Adding to this level of problematical abstraction is the fact that Alexander's formulation and evolution of his pattern theories was undertaken as much as a philosophical exploration as an engineering exercise. This means that the seminal literature does little to provide concrete bases for regularizing the notions of patterns in general or the set of patterns one might generate to describe a given domain of operations.

One way to circumscribe a working definition for a design pattern is to invoke the three most central elements of a pattern (dating back to Alexander's seminal formulations of the concept):

- *Context* - A setting or domain which serves as the coherent referential background to the circumscription of a problem and a solution. Both these other elements are described with reference to 'forces' which are discernible in the given context.

- *Problem* - A state or situation subsumed within the given context which can be described in terms of the interaction of 'forces' and which, as a whole, is considered to be something that should be mitigated or eliminated.

- *Solution* - A state, rule, product, or other form of intervention which once implemented will modify the interplay of forces involved in the stated problem and achieve an improvement in the context relative to the state in which that problem was initially specified.

These 3 elements and their interrelationships are summarily illustrated in Figure 31.

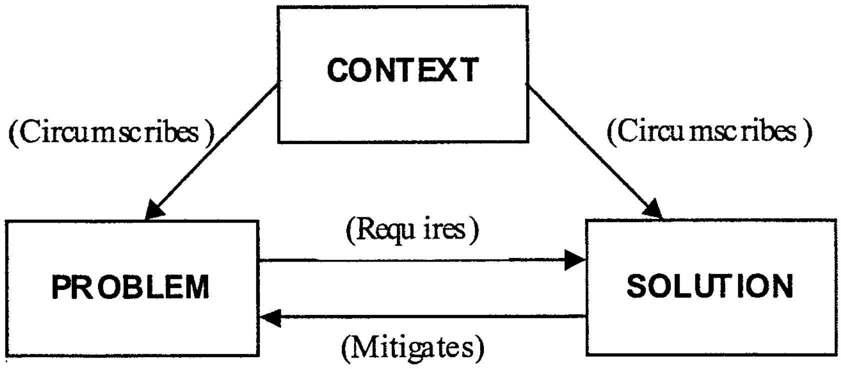


Figure 31: The 3 Primary Elements in a Design Pattern

The arrangement illustrated in Figure 31 is the minimal model for outlining a design pattern. This claim is based on the fact that in some fields and for some analysts the three elements depicted in Figure 31 are the only elements comprising their design pattern specifications. There are some pattern inventories in which these are either (a) the only elements represented in any of the entries or (b) the only elements which are consistently instantiated across all entries.

However, there are many more elements which have been put forward as necessary constituents of a complete design pattern specification. An enumeration of all these variants lies outside the scope of this report, and in any case it would not contribute to the expository core of the discussion.

To illustrate the range of elements that may (or may not) augment the 3 primary ones illustrated above, we provide Table 16. It contains a representative subset of elements often included in design specification templates. The relevance of the element set listed in Table 16 is that it is the set chosen as the basis of our analytical exercises during December 2004 and January 2005. In these exercises Terry Stanard and Randy Whitaker applied this set of elements to describe an example of one of our WCSS designs. The results of this illustrative analysis were briefed to the WIDE team in the January 2005 TIM.

Table 16: Working Subset of Available Design Pattern Elements

NAME	A concise and meaningful label for the pattern
PROBLEM	A statement of the problem to which the pattern is directed. The particular situation the pattern is being generated to overcome within the given context and forces.
CONTEXT	The setting within which the problem and its solution can be discerned to recur.
FORCES	A description of the relevant tensions between possibilities and constraints, how they interact/conflict with one another, and how they relate to the goals we wish to achieve (e.g., in relation to the problem).
SOLUTION	A specification for an intervention or outcome that resolves the problem in the given context. This may be outlined in the form of a product description, specifications, or new rules and procedures.
PICTURE / DIAGRAM	A summary graphic representation of the pattern and its components
VALIDITY	A measure or means for assessing the 'rightness' of the given pattern for the given context and problem
EXAMPLES	One or more sample applications of the pattern which illustrate how the pattern is or can be manifested to provide a solution to a problem in the given (or a closely analogous) context of operation.
SMALLER PATTERNS	Patterns which can be treated as subsidiary components of the given pattern.

As illustrated in Table 16, the three primary elements are allocated a label (Name). The 'Forces' which are invoked to describe the Problem and the Solution within the Context are given a category of their own. The Validity category relates to values, measures, and evaluations applied to the pattern. The remaining 3 categories (Picture, Examples, and Smaller Patterns) provide information for more richly illustrating the given pattern and interrelating it with other patterns evident in the subject matter domain.

To further illustrate the even wider range of elements or attributes which have been used as elements of a design pattern specification, we provide Table 17. This table lists a representative set of attributes or elements drawn from the literature on design patterns in the field of object-oriented programming. This list is not an exhaustive enumeration of the additional elements which have ever been invoked in design pattern applications, but it will suffice to portray how many more aspects are sometimes incorporated in describing a given design pattern.

Table 17: Illustrative Set of Additional Design Pattern Elements

RESULTING CONTEXT	The state or configuration of the setting after the pattern has been invoked or applied, including its consequences and prospects for ongoing problems.
RATIONALE	An explanatory justification for the pattern, ideally framed with respect to how it resolves its forces in appropriate ways.
RELATED PATTERNS	The static and dynamic relationships between this pattern and others within the same pattern language, system, or operational setting.
KNOWN USES	Describes known occurrences of this pattern (or its equivalents) and how these work in known settings or systems.
QUALITIES	Desirable attributes of the pattern. In the case of OOP, these attributes typically are taken to include:
• **Encapsulation and Abstraction**	Each pattern encapsulates a problem and an attendant solution in a particular domain of operations. By the same token, a pattern is an abstraction illustrating domain knowledge and experience that may apply at different levels of granularity within the domain.
• **Openness and Variability**	Each pattern should be open for extension or qualification by other patterns so that they may work together to solve a larger problem. Similarly, patterns should be capable of variation to fit additional or new circumstances.
• **Generativity and Composability**	A pattern generates a resulting context which can be evolved to progress toward the objective of an eventually complete overall solution. Patterns defined at a particular level of abstraction or granularity may be combined or integrated with other patterns at varying scales.
• **Equilibrium**	The manner in which the pattern balances the associated forces and constraints.

What is a User Interface Design Pattern?

For our purposes we will define a *user interface design pattern (UIDP)* as:

- a specific innovation or intervention (a Solution)
- realized as a set of features and functions allowing a user to access and control a specifiable aid or set of aids (Context)
- that addresses or mitigates a specifiable Problem
- where that Problem is framed with regard to actual or potential actions performed by a particular user (Context)

This definition qualifies both Problem and Solution with regard to a Context that must account for both the features of the stated Problem and the features of the identified Solution. It is sufficiently general to cover any situation involving the interaction of a user and his / her IT aids. Notice that under this definition the functionalities of the IT systems themselves are subsumed within the Context, and that the Solution is circumscribed in terms of gaining access to and / or control over those functionalities. For a UIDP to be 'work-centered', however, we must add the further qualification that the

Context and the Problem must be framed with respect to a specifiable task or work activity.

An Illustrative Example: The Mission Summary

Beginning with the earliest AFRL WCSS project for AMC (HISA), our team has consistently recommended the need for TACC users to have summary situation awareness (SA) over their pending workstream. We have prescribed multiple design concepts which display an ordered list of missions during their active case periods (i.e., during the entire process path leading from initial planning through mission execution). The first operational instantiation of this concept was the 'Sortie Palette' feature included as a subsidiary element of the GWM-WCSS during GAMAT Phase II. The Sortie Palette is illustrated as the highlighted area of the interface depicted in Figure 32.

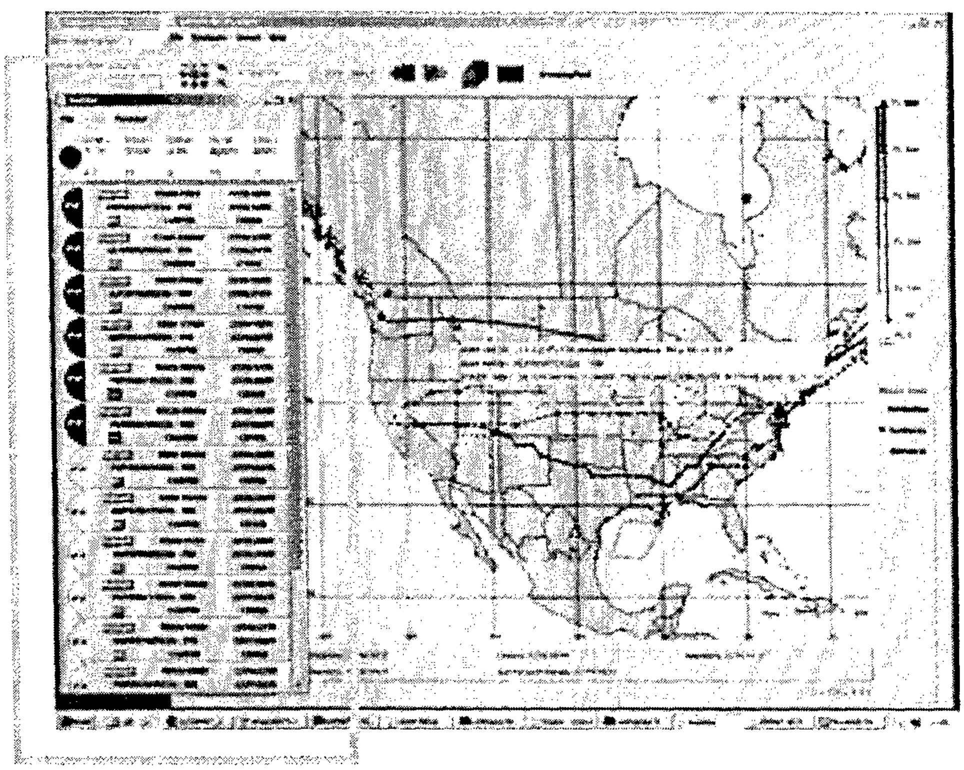


Figure 32: The Sortie Palette within the GWM-WCSS

As an exercise, we analyzed the Mission Summary WCSS concept in its most general form as an example of a design pattern. For the purposes of this illustration, we shall only go so far as to present an overview of the primary pattern elements (Context, Problem, Solution) as we see them relating to the Mission Summary. The first portion of this overview concerns the Context specifications we'd identified from our experience with recommending and designing multiple instantiations of such a workstream summary concept. We found that the Context portion of the canonical pattern needed to be qualified with respect to three levels of concern - the TACC organization as a whole, the work process conducted by TACC across multiple positions and roles, and the

perspective of an individual performing one of those roles. This three-way specification for the Context is illustrated in Table 18.

Table 18: Context Specs for the Mission Summary Example

Context: Organizational Scope	• Large organization (TACC) with many specialized positions • Positions include: Mission Planner, Barrel Master, DIP Shop, Flight Planner, Flight Manager, Execution Cell • All positions are working on the same class of work products
Context: Work Process Scope	• Each position organizes work around individual cases • Different positions may define their respective 'cases' differently (case is a flight for a flight planner vs. a diplomatic request for a DIP planner) • Each case is associated with a given mission • Large volume of cases (missions): 300 per day on average • Long timeline with each case (Up to 3 months lead time for planning) • Ideal work topology = linear feed-forward process: like a production line • Actual work topology = generally linear with an arbitrary number of cycles and loops in the course of the mission processing • Different roles manage different aspects of a case • These different roles must coordinate and collaborate to process a given mission
Context: Individual Scope	• Maintain awareness of current case load • Determine priority of cases • Anticipate handoffs to receive cases, send cases • Identifying problems with cases during planning cycle • Challenges: Large volume of cases, Long timeline associated with cases, and Routine or unmemorable nature of some cases (Channel missions)

Within this Context, we framed the Problem and Force specifications in accordance with the topics and issues noted in the relevant WCSS analyses and design efforts. These specifications are summarized in Table 19.

Table 19: Problems / Forces Specs in the Mission Summary Example

Problem	• Plan and monitor military aircraft (cargo) missions • Must manage complex work (case) features • Must balance conflicting priorities and demands • Planning and execution monitoring is distributed across positions with handoffs • Maintaining accurate SA over pending case workstream is difficult • Maintaining timely SA over pending case workstream is difficult • Significant cognitive and procedural burdens • High risk of information overload • High risk of errors and oversights in managing case workstream
Forces / Tensions	• Track large volume of cases vs. Inspect individual cases • Simple case index vs. Complete information on each case • Process cases on schedule vs. Process high priority cases first • Recall any one case over long period vs. Tendency to forget routine cases • Optimizing individual processing performance vs. optimizing collective team performance

Finally, we outlined the characteristics of the Solution (i.e., the Mission Summary in general form), as illustrated in Table 20.

Table 20: Solution Specs in the Mission Summary Example

Solution	• Summary listing of cases in workstream • Vertical stack of cases, ordered by time until (or since) launch • Concise format minimizes visual scanning • Selectable subset of all cases based on position • Selectable subset of all cases based on shift • Ready capability to index the set of pending cases (by mission ID) • Essential information on status • Link to additional visualization for more detail (expand, track on map) • Color coded alert status

How have Design Patterns been Applied in IT?

There have been 3 primary threads or streams of work in which the general notion of design patterns has been explored with regard to IT design and development. Each of these three streams of work is distinct with regard to the population of researchers involved, the field or discipline in which the work was conducted, and the level or aspect of IT products it can be considered to most directly address. All three, however, are relevant to the WCSS work that AFRL has been doing with AMC. In the following subsections we shall briefly review each in turn.

Design Patterns and the Form of Interface Artifacts: IADP

The first major application of Alexander's ideas in IT concerned the possibility of applying design patterns to describe the form of visible components in an information system (i.e., the 'interface', broadly defined). This line of work tended to concentrate on how the interface appeared or was structured. In this sense, this line of work could be pursued with primary attention to the artifact itself and minimal - if any - attention to the manner in which an actual user would engage or employ the artifact. In other words, the objectives of this line of work could be addressed strictly in terms of the visible interface product itself. Though much of the rationale for why this product was the way it was might well allude to what a user is expected to do with it, the expository focus is such that the user need not be in the picture.

This line of work typically focused on interpretation of user interface (UI) features in light of their being constituent components of a pattern language delineated with respect to the structure of the interface itself. This approach arose during the middle and latter parts of the 1980's, as graphic user interfaces (GUI's) began to appear on desktops in large numbers. A good example of this approach would be treating the desktop metaphor elements of the Macintosh interface (Apple Computer, 1992) both individually and

collectively as objects of design pattern analysis. The form of a 'window', for example, can be described in terms of its possible shape and size on-screen.

This artifact-centered approach to design patterns in IT was pursued by researchers and developers whose disciplinary affiliations could be categorized as including applied human-computer interaction (HCI), interface design, GUI design, and software engineering. In other words, practitioners adopting this perspective tended to be associated with design as it pertained to specific development projects or products. Collections of patterns associated with this perspective provide catalogs of the ways in which the interface can appear and function in and of itself. To distinguish this brand of design patterns from the others described below, we shall label them *interface artifact design patterns (IADP).*

Design Patterns and the Usage of an IT Artifact: IUDP

The second major application of Alexander's ideas in IT concerned the possibility of applying design patterns to describe the manner in which IT users interacted with the artifacts focused upon in the first application cited above and / or how such users employed these artifacts in their task activities. In this case, the focus was on the engagement between user and artifact, rather than on the features and configuration of the interface via which that engagement occurred. Though this perspective was predicated on the features of the given interface (and its underlying functionalities), analytical attention was directed primarily to the actions and possible actions the user could undertake with the interface (not the interface per se). Owing to this difference, this second line of work - though often intermingled with research and writings dedicated to the first approach above - cannot be properly subsumed under the artifact-oriented first approach. Though much of the rationale for why this product could be employed in a certain manner might well allude to features of the interface artifact(s) involved, the expository focus is such that the finer details of the product need not be in the picture.

This line of work typically focused on interpretation of user interface (UI) features in light of their representing a pattern language delineated with respect to what a user could do with the interface. This approach (as a variation on the first one) had a secondary status during the 1980's. However, it grew in importance as two phenomena emerged. The first was the proliferation of IT into more and more corners of the workplace. The second was the rise of the Internet as a medium through which work and commerce were conducted. Both these developments entailed a wider population of non-technical personnel employing IT in their everyday activities. As a result, more attention began to be paid to making IT products more 'user-friendly' to workers with little or no background in computer science or computer skills. It was also during this period that increasing attention was directed to user's work activities and requirements as the driving force in setting IT applications' specifications.

This line of work, applied to the World Wide Web and its interface artifacts, led to the assembly of pattern 'libraries' containing numerous examples of design patterns for

interactive online tasks and functions (e.g., van Welie, 2001; Laakso, 2003; Tidwell, 2000; 2002). The pattern specifications included in these collections typically describe both the task and the corresponding artifact in general terms, concentrating the descriptions on summary features rather than tangible specifics. To distinguish this brand of design patterns from the others described above and below, we shall label them *interface usage design patterns (IUDP).*

This usage- or activity-centered approach to design patterns in IT was pursued by researchers and developers whose disciplinary affiliations could be categorized as including theoretical human-computer interaction (HCI), cognitive engineering, and interaction design. In other words, practitioners adopting this perspective tended to be associated with design as it pertained to general aspects of worker or user experience or requirements (as opposed to the particulars of the artifacts these subjects employed). Collections of patterns associated with this perspective provide catalogs of the ways in which the interface relates to the needs and actions of the user(s).

Design Patterns in Software Engineering: SADP

The third major application of Alexander's ideas in IT concerned the possibility of applying design patterns to describe the form and / or function of the software modules which comprise the program(s) operating out of sight behind the user interface. It is this third line of design pattern work which has been developed to the most detailed extent, and it is this line of work which has resulted in the most widespread application. In the mid-1990's, software engineering researchers focused on the prospect of applying Alexander's design pattern concept to characterize modular software components - most particularly the 'objects' at the center of object-oriented programming (OOP). To distinguish this brand of design patterns from the others described above, we shall label them *software artifact design patterns (SADP).*

This idea 'caught fire' in the OOP community, leading to the publication of several books invoking design patterns as a means for analyzing and organizing object programming toolkits (Gamma *et al.*, 1995; Pree, 1995; Coplien & Schmidt, 1995; Gabriel, 1996). Out of this wave of publications, it was the 1995 book by Gamma, Helm, Johnson and Vlissides which had the most immediate impact. The influence of these four authors in steering OOP practitioners toward design patterns was so significant that they came to be collectively known as the 'Gang of Four'.[11]

The Gamma *et al.* book is almost entirely devoted to the notion of 'patterns' as they pertain to what these authors call 'micro-architectures' (atomic code composites also known as object structures). These micro-architectures subsume a set of static and dynamic relations among the objects (and/or their classes) that a programmer would ordinarily address in object-oriented development. These authors outlined a set of some

[11] This collective label has been treated as both positive and negative. Within the OOP community it is a respectful term for the ascribed progenitors of the design pattern movement. Among others more aligned with the other two approaches, the label is treated as a negative one (e.g., Tidwell, 1999).

23 patterns, which continue to serve as a fundamental reference set for design pattern inventories in OOP. Owing to the popularity of this book and its approach to defining and categorizing 'patterns' it is safe to say that this orientation most commonly encountered in the software development community.

Summarizing the Three Orientations to IT Design Patterns

There are 3 discernible streams of work in which the concept of design patterns has been applied in IT. There are no hard and fast boundaries between these 3 approaches and some researchers' work can be seen as overlapping at least 2 of the categories we've delineated. Still, it is constructive to illuminate the fact that not all IT design pattern work is addressing the same subject matter or, for that matter, addressing its subject matter from an orientation identical to other such work. A comparative summary of the three orientations is offered in Table 21.

Table 21: Comparative Summary of 3 IT Design Pattern Orientations

Pattern Type:	Interface Artifact	Interface Usage	Software Artifact
Acronym	IADP	IUDP	SADP
Focus (Specific)	Concrete elements of the interface product(s)	Actions and procedures the user can perform with the interface product(s)	The logical / functional components underlying the functionalities visible at the interface.
Focus (Figurative)	What is available for the user to see and to manipulate?	How the user engages (sees; manipulates) the UI in the course of his / her task.	The processing logic available to support what the user can see and do.
Background	• GUI research (1980's) • Software development (1980's and onward) • Web / HTML / Java development (1990's and onward)	• Human factors / performance studies • Cognitive engineering • Cognitive task analysis	• Software engineering • Software production management

Correlating the Three IT Design Pattern Orientations with WCD Methodology

In turning to our focal subjects of WCD and WCSS, we need to be able to correlate these 3 types of IT design patterns with the methodology by which we operate. The different orientations and their attendant priorities and foci can be mapped onto particular steps and phases in our standard WCD process path. Historically, we have characterized WCD in terms of two procedural outlines:

- *A Four-Step Procedural Path* - The earliest characterization of how we conduct WCD subdivided the work into 4 steps: work knowledge capture,

problem analysis, work aiding design, and work-centered evaluation. As noted in Chapter IV., a fifth step (the actual software development) needs to be added to complete the specification set.

- *A Two-Step Process Path* - The more recent characterization of how we conduct WCD subdivides the work into 2 general phases - problem analysis and design synthesis. The former encompasses work knowledge capture and problem analysis, while the latter encompasses work-aiding design, development, and work-centered evaluation.

A summary overview of how the 3 IT design pattern orientations correlate with our WCD procedural characterizations is provided in Table 22.

Table 22: Correlation Between IT Pattern Orientations and WCD Procedure

WCD Applicability:	Interface Artifact IADP	Interface Usage IUDP	Software Artifact SADP
By Step(s) In the development process path	• Work Aiding Design • Software Development • Work-Centered Evaluation	• Work Knowledge Capture • Problem Analysis • Work Aiding Design • Work-Centered Evaluation	• Software Development • Work-Centered Evaluation
By Phase(s) In the 2-phase process model	• Design Synthesis	• Problem Analysis • Design Synthesis	• Design Synthesis

As can be seen from Table 22, it is the IUDP stream of IT design pattern work which most comprehensively spans the range of procedural steps and phases by which we describe the manner in which WCD is conducted. The IADP orientation is most applicable once the WCD team's attention turns from problem analysis to design synthesis. This is also the point in the process path at which the SADP variant becomes most applicable. However, the IADP orientation is useful during the work-aiding design step, whereas the SADP orientation really doesn't come into play until work begins on constructing a WCSS prototype.

It would seem (based on Table 22) that of the 3 IT design pattern approaches or orientations, it would be the IUDP variant which would be most applicable to the manner in which we conduct WCD. This is consistent with the fact that the work-centered 'philosophy' prioritizes the activities by which actual workers conduct their work. It is therefore not surprising that the action focus of the IUDP approach would recommend itself as the best 'fit' for the activity focus of WCD. This is not to say that the IADP and SADP approaches are irrelevant or are to be discounted. However, both those other two approaches concentrate on the artifact being produced, and hence are the ones most applicable to all IT intervention methodologies, not just WCD. It is the IUDP approach, however, which most closely goes to the heart of the distinguishing features of WCD and WCSS (relative to conventional design and IT artifacts).

Evaluating the Practical Status of the Three IT Design Pattern Orientations

Based on these last points, it would seem that applying IT design pattern work in our WCD methodology would primarily be a matter of importing relevant resources and knowledge from the IUDP approach. This presumption, however, turns out to be problematical. The reason for this relates to the practical states of development for the 3 approaches. By 'practical states' we mean the status of each approach with regard to its maturity and / or robustness as something upon which we can based concrete design and development practices in our WCSS projects.

Even after our extensive literature search and review, we cannot claim to have attained absolutely complete situation awareness on the state of design pattern research as it pertains to IT applications. Still, the information we've accumulated to date is sufficient to provide a basis for a comparative evaluation of the 3 approaches as foundations for actual design and development practices. A summary overview of this evaluation is provided in Table 23.

Table 23: Comparative Evaluation of the 3 Approaches in Terms of Practice

Attributes of Practical Status:	Interface Artifact IADP	Interface Usage IUDP	Software Artifact SADP
• Coherence	Low	Very Low	High
• Specificity	Low - to -Medium (depending on example)	Low	High
• Available resources	Medium	Low-to-Medium	High
• Consistency across Resources	Medium	Low	High

Each of the 3 approaches was evaluated with respect to a set of 4 attributes. 'Coherence' refers to the cogency of the approach's models, frameworks, and conceptual bases. Coherence is necessary to provide a sound conceptual foundation for practical applications of the orientation. 'Specificity' refers to the level of detail to which the approach has demonstrated an ability to document design patterns as concrete specifications to be employed in an actual development project. Specificity is required to allow analysts, designers, and developers to share a working model of the WCSS being produced. 'Available resources' refers to the documented models and pattern libraries upon which a design team could draw at this point in time. Incorporating one or another approach within our WCD methodology would be facilitated to the extent that practical resources are already available for adoption. 'Consistency across resources' refers to the uniformity (of focus, of style, etc.) any such available resources exhibit. To provide a reliable foundation for design and development work, any orientation would need to maintain such consistency in the resources we attempted to use.

As Table 23 illustrates, the IT design pattern orientation previously cited as most relevant to WCD practices (IUDP) is also the one ranked lowest on all these 4 criteria. This approach's best 'score' (Low-to-Medium) was achieved with respect to 'Available

Resources'. This ranking was based on the availability of multiple design pattern sets best characterized as falling under the IUDP orientation. However, the abysmal rankings with regard to Coherence, Specificity, and Consistency make one wonder if the IUDP orientation can be made workable for actual project application. This issue will be pursued in the following sections.

A Closer Look at IUDP

The next step in our exploration of IT design patterns was a deeper review and analysis of the available design pattern resources in the IUDP category. This review will begin with examination of three sets of UI design patterns, then move on to a comparative analysis of them.

Three Illustrative UI Design Pattern Collections

There are three substantial libraries of UI design patterns that reasonably fall under the IUDP category. The first is the set of patterns compiled by Martijn van Welie (2001), which concentrate on features of the user interface and the actionable affordances provided the user. The second is the latest edition of a design pattern library compiled by Jenifer Tidwell of MIT (Tidwell, 2002). The third is a collection of patterns compiled by Sari Laakso of the University of Helsinki (2003). All three collections frame their contents with regard to tasks or actions on the part of the user. Some correlate at least a portion of their pattern entries with respect to features of the interface itself.

The first of these collections to be illustrated is that of Martijn van Welie. His collection is explicitly framed with respect to features of the user interface, but it is organized with respect to actions and tactics a user might perform in the course of a task. The collection - comprised of 26 pattern entries subdivided into 6 classes - is illustrated in Table 24.

Table 24: M. van Welie's Set of User Interface Design Patterns (from van Welie, 2001)

Modes / Selection / Physical Interaction	Guidance/Feedback / Navigation / Presentation
Modes	**Guidance/Feedback**
• Automatic Mode Switching	• Shield
• Helping Hands	• Hinting
• Mode Cursor	• Warning
	• Progress
Selection	• Undo
• Magnetism	
• Continuous Filter	**Navigation**
• Contextual Menu	• Wizard
• Focus!	• Softkeys
• Unambiguous Format	• Navigating Spaces
• Preview	• Container Navigation
• Setting Attributes	• List browser
• Command Area	
• Managing Favorites	**Presentation**
• Preferences	• Grid layout
Physical Interaction	
• Like in the real world...	
• Media Slot	

The second IUDP pattern collection is that of Jennifer Tidwell. It is illustrated in Table 25. This collection was first assembled under the label 'COMMON GROUND' circa 2000. A second edition appeared with a new website dedicated to design patterns circa 2002. Tidwell's stated goal is to generate a sample pattern language for human-computer interfaces - one which gives maximal attention to objectives and actions on the part of the user and minimal adherence to specific features of an interface artifact. As illustrated (Table 25), this collection consists of some 59 pattern entries subdivided into 8 classes.

Table 25: J. Tidwell's Set of User Interface Design Patterns (based on Tidwell: 2000, 2002)

Organizing the Content	Getting Input From Users	Direct Manipulation
• Overview Plus Detail • Hub and Spoke • Extras On Demand • Step-by-Step Instructions • One-Window Drilldown • Intriguing Branches • Multi-Level Help	• Good Defaults • Forgiving Format • Fill-in-the-Blanks • Input Hints • Input Prompt • Dropdown Chooser • Remembered Choices • Illustrated Choices	• Smart Selection • Edit-in-Place • One-Off Mode • Spring-Loaded Mode • Constrained Resize • Composite Selection • Simultaneous Views
Getting Around	**Showing Complex Data**	**Stylistic Elements**
• Clear Entry Points • Top-level Navigation • Color-Coded Divisions • Animated Transition • Detail View Navigation	• Sortable Table • Tree-Table • Alternating Row Colors • Cascading Lists • Jump to Item • New-Item Row	• Deep Background • Few Hues, Many Values • Contrasting Font Weights • Corner Treatments • One-Pixel Lines • Skins
Organizing the Page	**Commands and Actions**	
• Visual Framework • Center Stage • Titled Sections • Card Stack • Closable Panels • Movable Pieces • Progressive Disclosure • Progressive Enabling • Property Sheet • Diagonal Balance • Liquid Layout	• Multi-Level Undo • Smart Menu Items • Prominent Done • Prominent Cancel • Action Groups • Rollover Effects • Progress Indicator • Command History • Macros	

The third collection of UI design patterns is that of Sari Laakso (2003). This collection is somewhat different in stated orientation than the other two. Laakso based this collection primarily on those recurring design problems and / or bodies of general design knowledge that occur in UI design activities. This approach was claimed to align the resulting collection with the Goal-Derived Design (GDD) method that Laakso and others have been developing. The collection consists of 21 pattern entries organized under 6 classes. Perhaps ironically, at least half the classes and entries are framed with respect to specific features of the interface artifact (in contrast to the claim that they don't focus on the interface per se). A summary of the Laakso collection is offered in Table 26.

Table 26: S. Laakso's Set of User Interface Design Patterns (from Laakso, 2003)

Search • Continuous Filter • Continuous Highlight **Data Views** • Overview beside Detail • Expand in Context • Fisheye **Storage** • Rule Storage • Data Storage • Placeholder • Temporary Storage **Hierarchies and Sets** • Tree • Groups and Items	**Selecting and Manipulating Objects** • Double List • Editable Table • Pile of Items • Master and Instances **Time** • Calendar Strip • Schedule **Save and Undo** • Autosave • Global Undo • Object-Specific Undo • Deleted Data Storage

Even upon cursory inspection, the reader will notice that the three collections are distinct from each other. They vary in the number of specific entries (from 21 up to 59). They vary less in their number of constituent classes (6 to 8). However, even when considered in terms of their classes, there doesn't seem to be a lot of correspondence among the three sets. To evaluate the degree of correspondence the summary collection listings seem to have, a comparative review was conducted.

The variation in the number of specific entries was so great as to render such a comparison needlessly complex. As a result, the comparison was completed in terms of the subsidiary classes into which each of the collections is subdivided. This was considered a more illustrative approach for two reasons. First, in terms of relative size the three collections most closely match in number of classes rather than number of specific entries. Second, it is less cumbersome to try and evaluate the distinctions among the three collections at the class level of granularity. One way of sorting out and cross-correlating the three sets is illustrated in Table 27.

Table 27: Illustrative Comparison of the 3 UI Pattern Sets (van Welie, Tidwell, Laakso)

Theme or Topic	Van Welie (6 classes)	Tidwell (8 classes)	Laakso (6 classes)
Features of the UI (General)	• *MODES*	• *ORGANIZING THE PAGE* • *STYLISTIC ELEMENTS*	• *DATA VIEWS*
Visualization Format	• PRESENTATI ON • *SELECTION*	• *ORGANIZING THE CONTENT* • *ORGANIZING THE PAGE* • SHOWING COMPLEX DATA • *STYLISTIC ELEMENTS*	• *DATA VIEWS* • *HIERARCHIES AND SETS*
Data Selection & Filtering	• *SELECTION*	• *ORGANIZING THE CONTENT* • *ORGANIZING THE PAGE*	• *DATA VIEWS* • *SEARCH* • *SELECTING...*
Features of the Use Setting	• PHYSICAL INTERACTION	• *DIRECT MANIPULATION*	• *STORAGE*
Aiding and Feedback	• GUIDANCE / FEEDBACK	• *ORGANIZING THE CONTENT* • *GETTING INPUT ...* • *COMMANDS AND ACTIONS*	• *TIME* • *SEARCH*
Information Space Navigation	• NAVIGATIO N	• GETTING AROUND • *ORGANIZING THE CONTENT* • *ORGANIZING THE PAGE*	• *DATA VIEWS* • *HIERARCHIES AND SETS*
UI Control	• *MODES* • *SELECTION*	• *ORGANIZING THE PAGE* • *COMMANDS AND ACTIONS*	• *SELECTING...* • *SAVE & UNDO*
Work Input(s)		• *GETTING INPUT ...* • *DIRECT MANIPULATION*	• *SAVE & UNDO* • *STORAGE* • *SELECTING...*
Time Tracking			• *TIME*
Data Storage			• *STORAGE*
Aesthetics		• *STYLISTIC ELEMENTS*	

The 'Theme or Topic' column lists the general points or themes which were reflected in the pattern collections. At face value, some 11 such general themes were discerned. For each such theme, the class(es) from each collection containing at least one entry clearly associated with a theme is listed. Where the class includes entries that fit under more than one theme, the class is redundantly listed under both themes. For any class that is redundantly listed, its title is listed in italics.

As can be seen from the summary table, the three class sets don't align neatly. There are differences in the mappings from any one class set onto the set of themes. Clear disjunctions among the class sets are reflected in the fact that there is at least one identifiable theme which each set does not seem to address. The van Welie set doesn't clearly address work inputs, time tracking, data storage, or aesthetics. The Tidwell

collection doesn't clearly address time tracking or data storage. The Laakso collection doesn't seem to address aesthetics. There is no theme for which all 3 sets provide a non-redundant correlate.

Such an interpretive sorting is, of course, imprecise and subject to multiple interpretations. Even taking all that into account, it should be clear from this exercise that there is no straightforward correlation between any two, much less among all three, of these sets. This in turn is no surprise. The Tidwell and Laakso sets are claimed to have been assembled on the basis of precedent, not any comprehensive survey. Differences in the ways specific entries had been allocated to each set's respective classes explain a lot of the variations in topical coverage and the redundant entries.

It is also interesting to consider the themes or topics evident in our WCSS products to date which are nowhere addressed in the 3 collections. These include:

- Workstream situation awareness
- Selection of a given work unit from the composite workstream
- Work product output capabilities
- Any features geared to collaboration or coordination of one's actions with those of a colleague

Analysis: Issues in Applying Design Patterns to WCD and WCSS

In this section we shall summarize the major points that arose in our review and analysis of the design pattern concept and literature. These points have been selected on the basis of their representing particular issues or problems requiring resolution before design patterns can be reliably and repeatedly employed as the basis for WCD efforts.

The Problem of a Presumptive Knowledge Base from which to Draw

According to Alexander, design patterns cannot be discernible until there are many successful examples of them serving as evidence in the everyday world. This was no problem in his own work because of the domain (architecture) in which he worked. Architecture has been practiced for several thousand years, and architectural design patterns have evolved over that entire time span. In other words, the body of empirical design evidence available to Alexander was huge. Furthermore, the value of the designs Alexander was analyzing had been validated by daily usage over the millennia. For example, doorways have been stable and proven designs for longer than history records. There is little question that doorways have been refined to a stable form with little room for improvement (in their general characteristics).

Computer systems have been in widespread workplace use for only about a quarter century now. The process of developing and deploying workplace IT applications has been erratic in course, anomalous in outcomes, and driven as much by market

considerations as by the perceived quality of particular products. By and large the user interface has been the last thing to be prioritized in development. After 20 years we are still grappling with the desktop metaphor introduced broadly with the Apple Macintosh and then with Microsoft Windows. It is therefore not far-fetched to claim the introduction of windowing UI's represents the one and only major innovation in user interfaces since the days of textual command line protocols. On top of this is the fact that UI design and analysis is a recent field of scholarly research and an even more recent field of widespread applied practice.

This explains why there are so few UI design patterns represented in the collections available to date. The collections are relatively new, and all are explicitly claimed to be very tentative. None claim to be 'top-down' summaries of some global set of validated UI patterns, because such products are only recently introduced. Phrased another way, interface design has a long way to go before its body of empirical data can approximate even a fraction of the depth, breadth, evolutionary refinement, and degree of practical validation Alexander enjoyed in his original domain of architecture.

Descriptive versus Prescriptive Patterns

There are currently two ways of conceiving UI patterns:

- Patterns describing UI generalities observed in past solutions
- Patterns prescribing context-sensitive UI specifics in future solutions

As typically employed in diverse fields, patterns are usually *descriptive* in nature. In the case of UI patterns, they communicate features of a solution to a typical or generic user problem - e.g., data manipulation on desktop computers. Examples of such descriptive patterns are widely cited. Common capabilities of windowing environments such as sorting a list of files based on designated attributes or resizing a window to adjust on-screen space utilization are examples of such patterns. UI Patterns look back on what has already been achieved, to report what conventions have emerged. By definition, they are unconstrained by linkage to any particular domain of work. An air campaign planner and a computer help desk operator operate in very different work domains, yet both use such generic UI patterns in their day-to-day work. Domain independence has allowed UI researchers to survey a large population of examples to sift out those features such patterns represent.

In WCD, we need to generate a *prescriptive* solution to our clients' needs and requirements. We are custom designing an interface for specific people, and there are few, if any, UI solutions we can reliably copy to serve as such designs. To date, our WCSS products have been designed for a specific context of use and domain of practice. The value of a pattern lies not in how many application examples support it generally, but whether it demonstrably improves work performance in a specific work domain.

Owing to the fact that he was analyzing a vast body of practical knowledge and experience, Alexander's approach to formulating the design pattern concept was 'empirical'. More to the point, his approach was 'descriptive'. By this we mean architectural design patterns could be generated through describing what had been proven to work over a long period of time. Nobody expects a revolution in the number or type of basic architectural features. The body of data was already stable, and all that was required was for Alexander to sift through it and distill its generalities. Phrased another way, the constructs that Alexander originally laid out might be better termed 'designed patterns', because their representative specimens had already been designed again and again. In other words, the set of known Solutions was stable.

In UI design, we lack such a stable corpus of known Solutions. We are still exploring and trying out new things. We have yet to reach a point where we can reasonably claim to have laid out a working set of candidate Solutions, much less a validated set of optimal ones. As such, we are still interested in how to prescribe new Solutions. The original formulation of design patterns is ill-suited to such prescriptive purposes. This is well illustrated by the fact the canonical form of a pattern presumes a Solution as part of the input to the creation of a specification. In other words, the Solution is presumed to be a fixed point in the data. In our WCD work, the Solution is a variable we are trying to instantiate after achieving an understanding of the given Context and the Problem(s).

The Issue of the Importance to be Accorded the 'Context' Element

We find that we must put great emphasis on the 'Context' element. Context is a recurrent setting or background where a given design pattern is defined and within which it is applicable. Alexander's own early formulations of architectural design patterns emphasized the importance of context. He described an architectural design pattern in terms of a desired user interaction with the environment and architectural templates that achieve this interaction. Architectural design patterns include large scale items like a cathedral and bridge, and small scale like a sidewalk and a sitting window. Each of these patterns is meaningful in a particular context, but not others. This is why Alexander was careful to prioritize context in his specifications for what constituted a design pattern. A bridge is suited to the context of a river and an intersecting path of travel, but a bridge pattern is not suited to the context of a bedroom. Likewise, a window makes sense in the context of a bedroom, but not a foot path.

One of the problems with many treatments of Alexander's ideas is that they address Context, Problem, and Solution as three discrete elements. Though they represent different facets of the design scenario, they are not peer constructs. Both the Problem and the Solution are framed with regard to the Context. In other words, the Context must be specified so as to subsume the salient aspects of the Problem and Solution specifications. In particular, the Forces cited to describe the nature of the Problem and the manner in which the Solution resolves that Problem need to be framed in a manner consistent with and contained within the framing of the Context. All this is pretty clearly stated in Alexander's own work, yet many researchers seem to have lost sight of it. We

want to reiterate that the proper interrelationships among Context, Problem, and Solution are as illustrated in Figure 33.

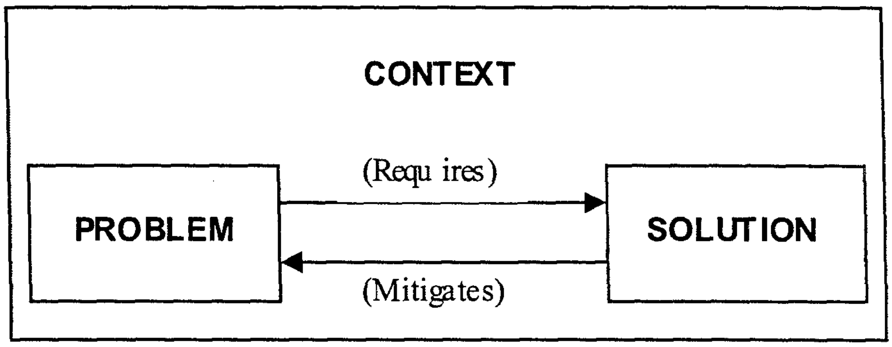


Figure 33: Our Model for the 3 Primary Design Pattern Elements

This point is of particular importance when discussing WCSS and the WCD process by which they are generated. The label 'work-centered' has since its inception connoted a prioritization of the immediate and situated circumstances of the work activities we seek to support with our design products. The early WCSS products were customized to meet the needs of one or another specific position within the TACC. As such, their features were matched to the specifics of a very well-circumscribed use scenario, not the generalities of all possible such applications.

Left with only the Context and Problem as their working bases, WCD practitioners have to be able to 'leverage' understanding of these two key elements in generating the third (the Solution). To do this requires that the Problem specification be as sound as possible. Because the Problem is framed with regard to the Context, this means the specification of the Context is the primary referential and analytical basis for effective WCD. In the following section we shall more closely examine the critical role of Context as a determinant of WCSS design applicability.

An Illustration of Contextual Variation: Design Patterns for Known WCSS

Now let us illustrate the criticality of Context with respect to two specific WCSS concepts drawn from our AFRL project work. The first - illustrated in Figure 34 - is the timeline tool concept generated in this project. The second - illustrated in Figure 35 - is the 'MOG Viewer' created in our inaugural WCSS project (HISA) 6 years ago.

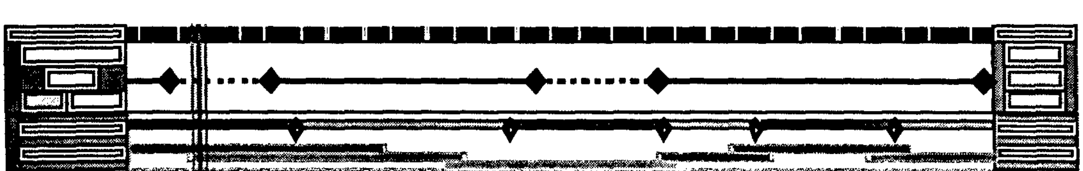


Figure 34: The Core Component of the Timeline Tool Concept

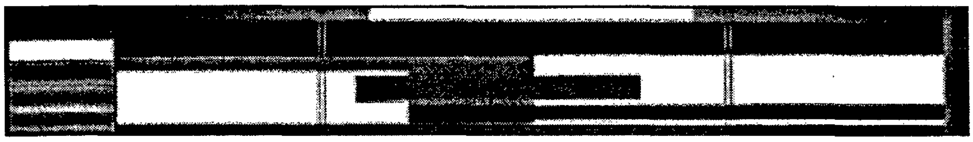


Figure 35: The MOG Viewer from the HISA Project

Both these WCSS concepts are based on a horizontally-delineated 'timeline' upon which missions or mission data are plotted. They therefore exhibit a 'family resemblance' which may or may not be indicative of an underlying design pattern. However, since each was developed at different times and for distinct applications, one could just as well treat them as specimens of distinct design patterns. In the remainder of this section we shall step through a series of design pattern specifications (equivalent to the treatment given the Mission Summary concept earlier). This will afford us a means for illustrating the extent to which these concepts can be considered alike or different based on different Context invoked in their respective design pattern specifications.

Case 1: The WIDE Timeline Tool

First let us consider the current WIDE timeline tool concept. Table 28 provides a Context summary for this concept's design pattern specification. Table 29 provides a summary of this pattern's specifications for Problem, Forces, and Solution (as represented by the timeline tool).

Table 28: Illustration of WIDE Timeline Tool Context

Context: Organizational Scope	• Large organization (TACC) with many specialized positions • All positions are working on the same class of work products (transport missions) • Different positions deal with different portions of a mission's subject matter • These different roles must coordinate and collaborate to process a given mission • Different positions utilize different IT resources, data, and visualizations
Context: Work Process Scope	• A successful mission is one in which a variety of resources are involved in a series of actions and events according to plan • There are many interrelated constraints among the mission factors for which these positions are jointly responsible • Mission execution is extremely time-critical
Context: Individual Scope	• Must maintain SA on a given mission as a whole • Must evaluate constraints holding among various factors involved • Must accomplish these things using a limited window on all mission data

Table 29: Illustration of WIDE Timeline Tool Problem and Solution

Problem	• Must manage complex work (case) features • Must balance conflicting priorities, demands, and conditions among the various mission features • Maintaining accurate understanding of a mission's status is difficult • Maintaining timely SA over a mission's status is difficult • Significant cognitive and procedural burdens involved in evaluating mission viability • High risk of information overload • High risk of errors and oversights in planning and monitoring a mission
Forces / Tensions	• One mission outcome vs. many factors determining outcome • Tractable set of mission factors vs. complex interactions among them • Optimizing individual processing performance vs. optimizing collective team performance • Tailored individual data visualizations vs. common information space
Solution	• Summary presentation of temporally-plotted mission data • Vertical stack of mission factors and events, correlated in a common timeframe • Concise format minimizes visual scanning • Visual cueing on problem states arising among mission factors and attendant constraints • Ready capability to index the set of mission factors • Essential information on status of mission as a set of coordinated events • Link to additional visualization for more detail (e.g., geo-spatial display) • Color coded alert status • Automated inference support for projecting ramifications and discovering alertable conditions

Case 2: The HISA MOG Viewer

Next let us consider the original HISA MOG Viewer concept. Table 30 provides a summary of all four elements (Context, Problem, Forces, and Solution) for this concept's design pattern specification.

Table 30: Illustration of the HISA MOG Viewer Pattern Specification

Context: Individual Scope	• Must maintain SA on a given mission as a sequence of traversals through airfields • Must evaluate constraints holding among airfield factors affecting mission plan viability • In this case, the focal factor of concern is maximum-on-ground (MOG) • Must accomplish these things using a limited window on all mission data
Problem	• Must manage complex work (case) features • Must balance conflicting priorities, demands, and conditions that could result in MOG • Maintaining accurate understanding of an airfield's traffic flow is difficult • Maintaining timely SA over an airfield's MOG status is difficult • Significant cognitive and procedural burdens involved in evaluating airfield availability with respect to MOG conditions • High risk of information overload in trying to predict MOG • High risk of errors and oversights in predicting and dealing with MOG
Forces / Tensions	• One mission itinerary vs. many itineraries resulting in MOG • Tractable set of mission factors vs. complex interactions among them • Massive textual mission data displays vs. demand for rapid visualization of airfield traffic flows
Solution	• Summary presentation of temporally-plotted traffic flow at a given airfield • Vertical stack of mission indicators, correlated in a common timeframe • Concise format minimizes visual scanning • Visual cueing on MOG states arising when too many aircraft are on-ground during a given period • Ready capability to index the set of missions / flights involved in MOG • Essential information on the projected period during which MOG will occur • Link to additional visualization for more detail (e.g., Form 59) • Color coded alert status • Automated inference support for projecting ramifications and discovering additional MOG conditions when evaluating alternative COA's

Notice that the specifications for the timeline tool and the MOG Viewer exhibit the following features:

- The MOG Viewer is reasonably portrayed on the basis of Context at the level of individual work responsibilities, whereas the timeline tool has to be framed with regard to a broader scope of Context up to the organizational level of granularity.

- With the exception of specific points relating to the subject matter being addressed in each of these two concepts, their Problem and Solution elements are quite similar.

Case 3: A Generic Temporal Visualization Display

Finally, let us generalize the examples of the MOG Viewer and the timeline tool into a generic temporal visualization aid. Table 31 provides a summary of all four elements (Context, Problem, Forces, and Solution) for a design pattern specification covering such a tool.

Table 31: Illustration of a Temporal Visualization Pattern Specification

Context: Work Process	• A successful case is constructively addressed as a series of actions and events • There are many interrelated constraints among the factors affecting case viability • Evaluation and decision making tasks are time-critical
Context: Individua l Scope	• Must maintain SA on a given case • Must evaluate temporal constraints holding among various factors involved • Must accomplish these things using a limited capacity for accessing and addressing all available data
Problem	• Must manage complex work (case) features • Must balance conflicting priorities, demands, and conditions among the various case features • Maintaining accurate understanding of a case's status is difficult • Maintaining timely SA over a case's status is difficult • Significant cognitive and procedural burdens involved in evaluating case viability • High risk of information overload • High risk of errors and oversights in processing a case
Forces / Tensions	• One case subject matter outcome vs. many factors determining outcome • Tractable set of case factors vs. complex interactions among them • Optimizing individual processing performance vs. optimizing collective team performance • Tailored individual data visualizations vs. common information space
Solution	• Summary presentation of temporally-plotted case data • Vertical stack of case factors and events, correlated in a common timeframe • Concise format minimizes visual scanning • Visual cueing on problem states arising among case factors and attendant constraints • Ready capability to index the set of case factors • Essential information on status of case subject matter as a set of coordinated events • Link to additional visualization for more detail • Color coded alert status • Automated inference support for projecting ramifications and discovering alertable conditions

All three of these examples result in the specification of a Solution based on a temporal display. The primary differences among them are the specificity of the subject matter or data being handled and the level of granularity for the Context specification required to describe each of them. A number of issues and questions are illustrated by this set of closely-related examples, including:

- Do these represent 3 distinct 'design patterns', or one general one with 2 specific variations?

- At which level of specificity and granularity would any or all of these 3 candidate patterns be reasonably fit for accretion to a pattern library?

- Isn't it the case that pinpointing the precise Context specification (e.g., in terms of granularity) is the key to generating, compiling, and evaluating design patterns for a given work environment?

It is this last question which is most pertinent to this discussion, owing to its practical ramifications. In the three examples above, the Problem and the Solution elements are relatively uniform across the different cases. What distinctions can be drawn among them are tied to variation in the Context description. This implies that coherence and consistency in a design pattern repertoire will be most effectively obtained by improving the quality of the Context specifications being developed. In the following section we shall conclude with an examination of some candidate approaches for improving the state of Context specification.

Pursuing More Effective Context Specifications

In the final period of the WIDE 6.2 UI design patterns work, we turned our attention to the issue outlined in the previous sections - i.e., what might be done to make design pattern practices more tractable for our WCD purposes through better Context specification. We can nominate two candidate approaches to devising better such specifications - both of which have been initiated within AFRL/HECS. The first is an outline for the set of factors relevant to the portrayal of Context, and the second is a model for the 'work ecology' often cited as a key construct in WCD.

Candidate Approach 1: A Descriptive Framework for Context Factors

The first candidate approach was generated by Terry Stanard and Randy Whitaker during the final phase of this study. In effect, we attempted to outline the set of possible features and factors descriptive of the work activity domain and work environment which were either evident in our prior WCSS projects and / or recommended on theoretical grounds. This set of descriptors can be subdivided into two primary parts. The first consists of features descriptive of the work in terms of objects, relationships, and attributes. The second consists of those features specifically related to the conduct of the target work activity.

The draft framework for the first of these components is illustrated in Table 32. At the highest level of reference, the target work is characterized in terms of a domain of operations. This construct represents the operational setting in which the work is conducted. For each such domain of operations, there will be a set of roles and functions.

These are the categories onto which particular people, teams, and organizational units will be assigned. For each of these categories, two classes of descriptive information will be compiled. The first is information descriptive of the general properties of the work being performed by that role or function. The second consists of an inventory of the relevant properties of the work setting.

Table 32: Descriptors of the Work Domain and Work Environment

Domain of Operations	Description of the overarching operational context for the work being analyzed. For each such operational domain specification, there will be ...	
Target Roles & Functions	Descriptions of the functional roles and functions evident in the operational domain. For each such role's work activity, there will be ...	
Properties of the Work Performed Specific features and factors peculiar to the work activity of a given actor / role, such as...	• Work objectives • Work products • Units of work • Work process • Linear or non-linear, handoffs, dependencies, timeline • Volume of work	• Work flow patterns, peaks • Typical user and range of users • Work management strategies, including those imposed by the organization, technology, personal habits
Properties of the Work Setting	Descriptions of those factors and features evidenced by the work environment	
Physical Factors	- heat - lighting - noise levels - vibration	- space constraints - ventilation - relevant physical processes - physical constraints
Organizational/ Political Factors	- lines of authority - norms and values	- rules, policies - reward systems (economic, informal)
Social Factors	- cultural features and attributes - informal power structures - trust relationships	- local experts - peer networks
Technology / Resource Factors	- resource types - capabilities & capacities - lines of accessibility and constraints thereon	- personal equipment - requisite expertise - health and safety factors

The second major component of this framework addresses the work activities themselves. These activities are addressed as being related to one or more of three general classes of action:

- *Behavior* - Physical or instrumental procedures performed in the work setting. This category includes those observable actions and activities associated with the work domain.

- *Cognition* - Those actions and / or activities conducted mentally or perceptually, such as decision making. This category includes non-observable

cognitive phenomena as well as those tasks whose outcomes and products are predicated on mentation.

- *Coordination* - Those actions and / or activities dedicated to achieving collective ends by effectively relating (e.g., synchronizing) one's own cognition and behavior with respect to features of others' participation in the domain of operations.

Certain features - e.g., triggers, constraints, timing factors, and challenges - are applicable in each of these 3 categories. A summary combined listing of examples drawn from these three categories is provided in the two tables below. Table 33 gives an illustrative comparison among the categories in terms of their respective actors, critical events, key tasks, and information requirements.

Table 33: Key Elements in Three Categories of Work Activity

Work Activity		
Coordination	**Cognition**	**Behavior**
Set of Actors (e.g., Unit, Team)	Actor (Mental / Cognitive Activity)	Actor (Physical / Instrumental Activity)
Coordination points	Critical decision points	Critical events
Situated coordination tasks	Situated cognitive tasks	Situated behavioral tasks, routines
Critical shared information	Critical individual information	Critical data and criteria

As Table 33 illustrates, these key elements in work activity are qualified with the particular manner in which they are engaged in relation to their respective activity types. These criteria only provide a basic description of the types of activity which might be subsumed under one or another of the categories. These may be diverse and numerous. Some illustrative listings of representative activities for all three categories are provided in Table 34.

Table 34: Examples of Activities Associated with the Three Categories

Work Activity		
Classes of Coordination	**Classes of Cognition**	**Classes of Behavior**
• Sharing information	• Monitoring	• Control tasks
• Consensus building on subject matter status	• Situation awareness (SA)	• Organizing objects
• Distributed Decision making	• Decision making	• Constructing, assembling
• Distributed Planning	• Planning	• Material movement, transport
• Communicating intent	• Forecasting	• Skilled performing
• Motivating and Inspiring Others	• Event, object, or pattern detection and recognition	• Generating documentation
• Goal Setting	• Hypothesizing	• Speaking / briefing
• Delegation	• Perceptual judgments	• Reading
• Handoffs	• Diagnosis	• Operating equipment
• Brainstorming	• Inspection	• Communicative behaviors
• Relationship building	• Problem solving	• Checking routines
• Negotiation	• Course of action analysis	
• Conflict resolution	• Mental simulation	
	• Managing Time	

Candidate Approach 2: Eggleston's Work Ecology Model

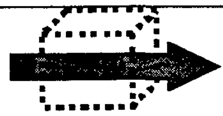
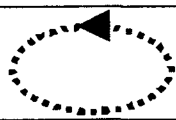
The second candidate approach to better Context specification also originates within AFRL/HECS. It is the 'work ecology model' under development by Dr. Robert Eggleston (cf. Eggleston, 2005). The term 'work ecology' has long been employed to connote the "... intrinsic milieu of the work organization (e.g. basic environment; type of product(s) development; processes)" which provides the setting for WCD analysis and design. As such, a viable model of such a work ecology would recommend itself for linking UI design pattern innovations to AFRL's extant WCD approach and methodology.

A work ecology is a composite concept. It covers a variety of aspects of the workplace, such as the social, cognitive, information, data, and physical. An ability to generate a coherent model interrelating these disparate factors would allow a WCD team to:

- Reveal the structural basis of the target work as practiced
- Reveal the structural basis of problems evident in the workplace
- Reveal the work phenomenon from the perspective of both the organization and the individual worker(s)
- Better and more efficiently understand the essential characteristics of the target work

There are three minimum components of a work ecology specification in this approach. They are summarized in Table 35 below.

Table 35: The Work Ecology Model's 3 Minimum Components

	*Product-Based Environment* A work setting framed at the highest level of generality with regard to the generation of specifiable work products.
	*Eco-Systems* Networks of resources and processes which constitute discrete subunits of the product-based environment
	*Agents* Actors whose functions contribute to or affect the operation of the eco-systems and hence the product-based environment

Working with these basics, Eggleston has been able to generate a notional work ecology model for the TACC mission planning and execution process path. This model is illustrated in Figure 36.

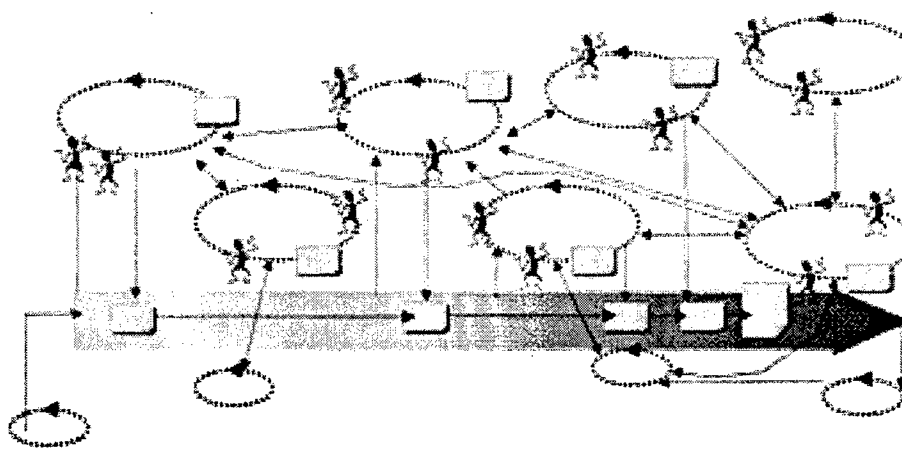


Figure 36: A Work Ecology Model for TACC Mission Operations

Having identified improved Context specification as the key to making UI design patterns a robust adjunct to WCD, we have gone on to identify two candidate bases for pursuing this objective. Both the candidates are products of AFRL/HECS. Both are in early stages of formulation and refinement. The development and application of either or both of these candidate frameworks to provide a firmer foundation for generating and compiling UI design patterns must, however, be left to future research.

Summary and Conclusions

The concept of a 'design pattern' holds much promise for AFRL's WCD methodology and WCSS product development projects. This concept has been under continual theoretical development and recurrent practical application for over decades now. The volume of literature and research dedicated to this subject give fair indication of its perceived value.

All this notwithstanding, design patterns are not yet an off-the-shelf remedy for better documenting and organizing the UI innovations we've been producing. There are specific points we've identified where design pattern theory is either less than fully mature or in need of refinement to meet our needs.

To contribute to improved WCD and WCSS practices, design patterns will have to be developed that are more specifically geared to the work-centric and context-sensitive character of our target applications. In particular, design pattern models and specifications will need to be made more coherent and consistent before they can reliably support prescriptive rather than simply descriptive ends. We have identified the most critical leverage point in achieving this - a more robust capability for defining the canonical Context element with respect to which all else in the design pattern schema is qualified. Finally, we have identified two emergent AFRL/HECS frameworks which hold promise as candidate bases for generating more robust Context specifications.

CHAPTER VI.
Advanced Information Composition

Background

One of the hallmarks of the work-centered support system (WCSS) prototypes developed to date is the ability to draw on multiple and disparate data resources and distill the data obtained into a form, format, and presentation tailored to the context of specific work activities, work demands, and work objectives. If collating and presenting data were all there was to providing effective task support, work-centered design (WCD) would boil down to an exercise in data management. This simplistic interpretation would greatly underestimate both the scope and the complexity of the issues addressed in the WCSS concept.

The reason for this is that 'information' is not properly construed as being identical with 'data'. A datum need be no more than a perceived difference or distinction in a medium - i.e., one 'bit' construed against a referential or perceptual background. Information, however, requires more than mere perception or recognition. As the classical specification puts it, "information = data + meaning". It is the meaningfulness of a data artifact - its semantics - which constitutes its status as 'informative'. This is the basis for Gregory Bateson's elegant definition of information as 'any difference that makes a difference'. (Bateson, 1987)

You can be provided all the data relevant to your task, but if it is presented in a foreign language or coding scheme you do not comprehend, it is just so much gibberish. You are not 'informed' until and unless you 'make something' of the data. As such, there are two issues involved in information technology support. These can be correlated with two basic questions with which any IT user must tacitly grapple on a continuous basis:

- *What (do I have to work with)?* This question addresses the availability, form, and state of the data the user confronts. Every time the user looks at his / her user interface (UI) display, the 'What?' question is being posed anew.

- *What does it mean to me?* This question presumes an answer to the first (i.e., a survey of the basic 'what') and extends to the interpretation of basic data, correlation with the subject matter at hand, and any derivative or projective inferences made upon it.

Providing all the possible data and expecting the user to 'connect the dots' as he / she deems fit is a poor tactic for assuring the data presentation is 'meaningful' and hence 'informative'. Indeed, such an approach is practically guaranteed to degrade, rather than enhance, task performance in most cases. Even in cases where such a shotgun approach affords the user everything required to perform his / her decision making task(s), it does so at the expense of the user. The effort required to 'connect the dots' or make sense of

the data (i.e., to distill task-specific 'meaning') detracts from the user's available resources and time. This is so well-known a problem that the human factors community has long given it a specific label - *cognitive burden* or *cognitive overload.* Ironically, this situation is colloquially known as 'information overload' - a term that generally refers to an excess of basic data, and not information in the strict sense.

Our WCSS can be characterized as composing optimally work-centered 'information' out of a potentially vast range of back-end 'data'. However, a work-centered application is not just understandable - it must be meaningful in the context of the given tasks and work activities. In other words, the value-adding 'meaning' induced through the manner in which a WCSS blends and presents data is 'meaningfulness' with respect to the work being supported. Through their emphasis on presenting users not just the 'what' (i.e., raw data) but also 'what it means to me in doing this task', WCSS enhance both situation awareness and problem understanding.

Two Perspectives on the Subject of Information Composition in WCSS

Effective such 'information composition' thus exemplifies our WCSS work. At this point in time, our team has accumulated enough experience and products to begin examining how such information composition has been demonstrably done well and how it could best be done in the context of WCD. Clues to the bases for a methodology for composing work-centered information visualizations from various data sources. The prospects for and form of such a methodology will be investigated from two perspectives:

- a *technical perspective* addressing the mechanics of retrieving and fusing data to support WCSS (e.g., via agents; via middleware; via data synthesis techniques) and

- a *use perspective* addressing how the fused data is best presented so as to constitute optimized 'information' supporting work activity.

Some of the main factors distinguishing the technical from the use perspective (and vice versa) are summarized in Table 36. As the table indicates, the two perspectives are intrinsically interwoven. The support capabilities devised with respect to the technical perspective afford the foundation for crafting displays in accordance with the use perspective. In turn, the benefits of these displays (from the use perspective) justify the effort invested in maximizing and optimizing the background support provided from the technical perspective.

Table 36: Two Perspectives on Information Composition in WCSS

	TECHNICAL PERSPECTIVE	USE PERSPECTIVE
FOCUS	The mechanics of providing data to the WCSS interface(s).	The means for optimally 'informing' the user based on the data available.
SUBJECT MATTER	Data resources, means of access, formats, protocols, and fusion	Work information requirements, cognitive demands, etc.
DESIGN FOR...	'What lies behind' the visible WCSS UI's - e.g., back end data resources, intermediate elements	What's provided on the visible WCSS UI's - visualizations, action opportunities, etc.

Both perspectives are intrinsic to devising effective WCSS products, and each is complementary to the other toward achieving these ends. The technical perspective is important to our WCSS project team at this point because:

- UI designs are just 'pie in the sky' unless you can populate them with actual data.
- Accessing and processing data from AMC legacy systems and other sources has been a major issue in each of our WCSS development projects.
- Technical issues relating to supportive data access, etc., have often been at least as complex as those relating to the WCSS UI itself.
- Now that the AFRL WCSS team is operating as part of the TACC ATD process, we are obligated to 'toe the line' with respect to official protocols and real-world constraints.

Conversely, the use perspective is important because:

- Our WCSS to date have 'sold themselves' based on their immediate and visible operational merits.
- These merits have largely related to affording users the ability to visualize situations, constraints, and opportunities in a manner fitting their work practices, demands, subject matter.
- In other words, one of the hallmarks of our WCSS has been our ability to effectively 'compose' data so as to optimally 'inform' the user.

How This Discussion will Proceed

The results of the WIDE 6.2 research on the technical perspective will be summarized with respect to current conditions and emerging opportunities related to methods for:

- defining and generating intermediate data representations
- specifying such intermediate representations and
- specifying intermediate technology support elements (e.g., middleware, fuselets and meta-data).

This will be framed with specific regard to the AMC / TACC work currently ongoing (under the aegis of the WIDE 6.3 project and the timeline tool that is its central product during this reporting period).

The use perspective will explore the prospects for defining a methodology for composing work-centered visualizations in such a way that both situation awareness and problem understanding are enhanced. This in turn requires that we give attention to any visualization tactics and techniques which contribute to the following objectives;

- increasing available perceptual and memory resources
- reducing search time and effort such as grouping data that are used together
- representing a large amount of data in a manageable workspace
- spatially indexing data in the context of UI layouts, and
- exploiting relevant data transformations and organizing data in otherwise meaningful ways.

This will be framed with specific regard to the AMC / TACC work currently ongoing (under the aegis of the WIDE 6.3 project and the timeline tool that is its central product during this reporting period). In addition, the central expository background will be AFRL WCD and WCSS work and products over the last 6 years.

PART 1: THE TECHNICAL PERSPECTIVE[12]

The Problem

A key requirement of work-centered support systems is the ability of the system to bring together data from multiple data sources. In fact, each of the work-centered systems the WIDE design team has been involved in implementing in the last few years (GAMAT,

[12] This portion of the Advanced Information Composition chapter was submitted in a paper entitled 'Information Composition for Work-Centered Support Systems', BBNT Solutions LLC, March 2005.

HISA, and now WSRD and WIDE) have included this as one of their prime selling points to end users: for the first time the users would have the ability to bring together disparate information sources onto one screen (or one map, or one timeline) that he had never before been able to correlate in a single view. As critical as this attribute of the graphical user interfaces have been to these systems, it's been even more critical to the software system design. With each of these systems, the software design to bring data from multiple sources into the system and allow it to be integrated has been one of the most expensive tasks. This paper begins to explore some of the issues around designing software to assist in information composition for work-centered support systems.

The Information Mediator Solution

Figure 37 shows a block diagram of a typical work-centered support system, following the prototypical work-centered support system described in our paper on Evolvable Work-Centered Systems. Our prototypical work-centered system consists of three main components. First is the Data Acquisition Module - the component of the system that is responsible for acquisition, decoding, and storage of data in which any users may have an interest. Second is the Analysis Module. In the Analysis Module, typically some combination of rule-based and algorithmic code, the raw data acquired by the Data Acquisition Module is filtered and transformed into higher-quality information ("decision-quality information") of immediate interest to the user. For example, in the WCSS-GWM, the Analysis Module identifies particular upper-air turbulence observations that threaten the successful operation of air missions. Finally is the Presentation Module, which includes the Graphical User Interface (GUI) with which the user interacts directly as well as reasoning algorithms which try to prioritize information for viewing by the user.

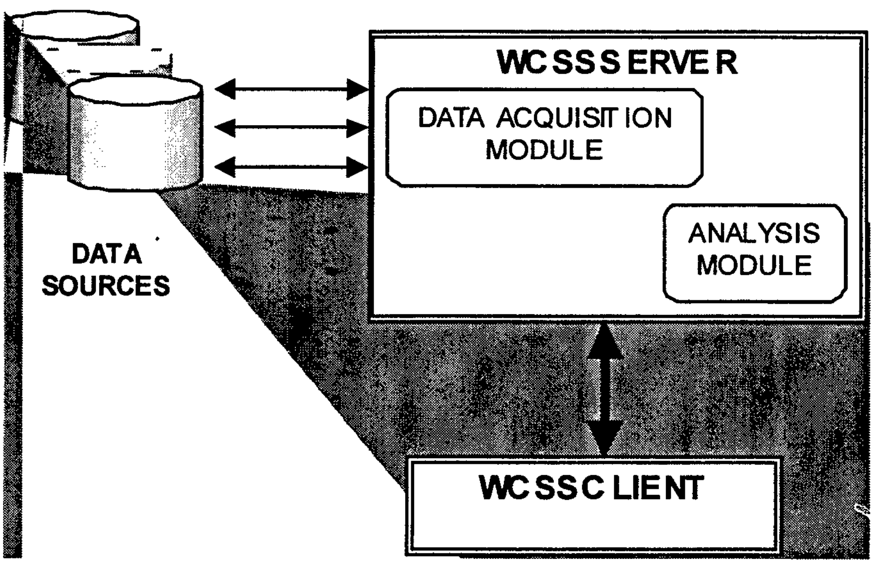


Figure 37: Prototypical Work-Centered Support System Architecture

The solution we propose here is a refactoring of the architecture presented in Figure 37. We pull out the Data Acquisition Module of the prototypical work-centered support

system, into a reusable Information Mediator component. The Information Mediator satisfies most of the same requirements as the old Data Acquisition Module and is responsible for acquisition of data and reformatting it into standard format. The difference is that the Information Mediator is designed with the intention of it being a stand-alone service available to multiple work-centered support system servers (or other information subscribers). The Information Mediator is intended to implement web services that can be used by subscriber applications to get information in standard format XML files. A block diagram of the proposed Information Mediator solution is presented in Figure 38.

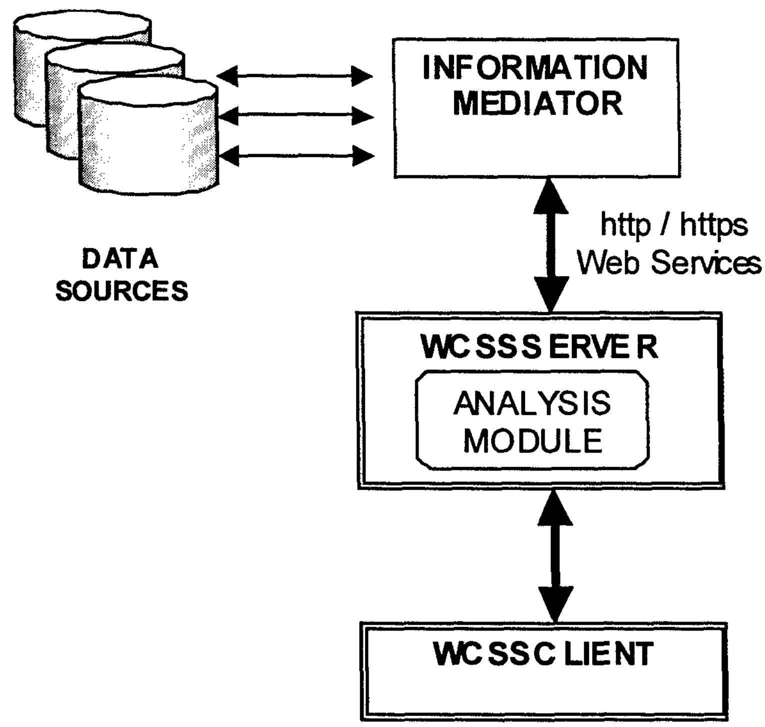


Figure 38: Information Mediator Architecture Solution

Requirements for an Information Mediator

The information mediator we define here will be a software module that will sit between the various data sources needed by the WCSS on one side, and the WCSS server on the other side. All information needed by the WCSS will come through the information mediator. We anticipate the information mediator being ultimately accessed through a set of web services, although the underlying implementation may very well include a publish-subscribe component such as JBI.

- The Information Mediator solution must provide a standard way to represent the information needs of the work-centered support system. As an initial pass at defining the representation, we use a standard XML schema. This will not suffice in the long run; the representation will eventually need to contain

significant semantic information which cannot be represented directly in a simple XML schema.

- The information mediator solution must produce a standard output format that matches the information needs of the WCSS. Again, as an initial pass, the information mediator will produce XML files corresponding to the XML schema used to represent the information needs.

- The information mediator must be able to provide both static and real-time data. In practice this means there must be multiple services provided. One set of services will be used to provide static data; for example, a service would provide information about military airfields around the world – latitude, longitude, ICAO code, runway information, standard operating hours, etc. A WCSS server would, upon initialization, query the information mediator once for this static data. Data that changes on a real-time basis; AMC mission data, for example, would be provided by services that are more geared to provide changes in mission data received over the last few minutes. Note that these services actually mimic the various RIDL reports used by GAMAT over the last few years.

- The information mediator must be able to access data from a variety of sources. In practice this means that the information mediator contains a library of standard data exchange and reformatting modules. The information mediator should be able to acquire data by ftp, http(s), or direct database queries. Each raw data source needs to be understood to the point where it can be mapped to WCSS information needs and reformatted – in other words, to where it can be recast in a standard XML schema. In our initial cases, this is fairly easy, although this job will get harder as we include more information sources. An example of a more complicated information mapping problem taken from a slightly different domain: Suppose we were designing a work-centered support system to deal with the in-transit visibility problem. This system would be responsible for being able to show where in the world military shipments are at any time. There are multiple information sources here, which generally use different primary keys to denote the items being shipped. In some cases, the items being shipped would be labeled by NSN (National Stock Number, a 13-character key). In some cases, items would be identified by NIIN (National Item Identification Number – a substring of the NSN) or LIN (Line Item Number – an entirely different indexing scheme); in other cases items might be listed only by a text description. To make allow all this information to be connected in a work-centered support system would require normalizing all these different types of data – a task which could range from difficult to impossible.

- As an eventual requirement, we would like the information mediator to be able to begin to accept information from new data sources on the fly. This

would be the basic mechanism by which our work-centered support systems will someday be able to integrate data from novel sources.

Examples Taken From the WIDE Timeline Effort

This section presents some examples of the information mediator concept as it would be used for the WIDE Timeline development effort.

We start by presenting a simplified representation of the information needs of the timeline. For ease in understanding we use a very simplified XML-like schema here, as opposed to a full XML schema (which while easily machine-readable, is not necessarily pleasant reading).

The WIDE timeline will basically consist of representing information about missions, airfields, aircraft, and aircrews.

Information needs for the timeline:

```
<mission>
        <plan>
                <schedule>
        <actual>
                <schedule>
        <revised plan>
                <schedule>
        <origin-airfield>
        <destination-airfield>
        <alternate-airfields>
        <aircrew>
        <tail>
        <dip-clearances>
        <pprs>
```

where a schedule looks like:

```
<schedule>
        <etd>
        <eta>
        <flight-plan>
                <waypoints>
```

and an airfield looks like:

```
<airfield>
        <icao-code>
        <name>
        <latitude>
```

<longitude>
<operating-hours>
<quiet-hours>

and an aircraft looks like:
<aircraft>
<tail-number>
<aircraft-type>
<configuration>

and an aircrew looks like:
<aircrew>
<aircrew-identifier>
<aircrew-type>
<certifications>

Note: The flight-plan part of this schema will eventually be in a standard CRD (Common Route Definition) format.

Data Sources Available to the Information Mediator

IFM Database

The IFM Database (currently used by IMT in the TACC) contains much of the mission data needed by the timeline. The mapping between the IFM database tables and information needs is quite complicated, and will perhaps be detailed in a further iteration of this paper. In general terms, though, mission identification, itinerary, and schedule information are all contained in the IMT_DASHBOARD table. Flight plans are complicated to piece together in this database – individual waypoints are split between the FLT_PLN_PT table (which contains the airport_id or air-route-segment-waypoint id for each waypoint of a flight plan) and the FLT_PLN_PNT_EVT table (which contains the time of each waypoint). The airport_id or air-route-segment-waypoint id then needs to be looked up in either the ARPT (for airport_id) or AIR_RTE_SEG_WAYPT table to recover a LOC_ID, which can then be looked up in the PT table to recover a latitude and longitude. It should be noted that even being able to define a syntax for an input to the information mediator that would define this data transformation (from input IFM database to output XML containing flight plans) would be a challenging problem.

In the IFM database, position reports (which would be expected by the timeline in the 'actuals' portions of the mission schedule) are contained in the IMT_LEGS_FLIGHT_DATA table.

CAMPS Database

The CAMPS database contains mission schedule data, information about airfields and aircraft characteristics (including cruising speeds, typical on-ground times, crew duty day limitations by aircraft type).

GDSS2 Database

The GDSS2 database contains all the data contained in the IFM database (as well as much other data besides). The GDSS2 database is entirely restructured, though. We do not yet understand the mapping between GDSS2 database tables and the information needs of the timeline, although the mapping does appear to be simpler than the mapping between IFM database tables and the timeline. In particular, the GDSS2 database contains several 'performance tables', which are non-normalized tables, containing data joined from other tables, put together for the purpose of optimizing certain queries. The PT_AM and PT_SRT tables together contain much of the information the timeline will need about air missions (PT_AM) and sorties (PT_AM). Itinerary information is contained in the PT_AM_ITIN table, and basic flight plan information is contained in the PT_CFP_WAYPNT table – no such multi-table joins will be needed here like were needed in the IFM database.

PART 2: THE USE PERSPECTIVE

The balance of this chapter will discuss the 'use perspective' on advanced information composition. This discussion will proceed with particular focus on the WCD methodology and the WCSS products generated in the course of AFRL's projects for Air Mobility Command starting in 1999 and continuing through the present reporting period.

A Review of 'Information Composition': Historical Background

The term 'information composition' and closely-allied terminology appear throughout the history of information technology (IT). This phase is used in a colloquial sense to connote 'the assembly / organization / arrangement of information'. In this sense such usage of the term 'composition' extends back earlier than the arrival and proliferation of automated information systems. For example, the process of organizing and laying out a newspaper or magazine page prior to printing has long been called 'composing' or 'compositing'. The static character of an 'information composition' connoted in the field of printing has, however, carried over into the computer age. The US Army still employs the term 'information composition' to mean the process of organizing material in preparation for printing (*FM 11-43*, Section 1-1). In recent research articles on tools for integrating access to heterogeneous databases, the term 'information composition' has explicitly been taken to mean no more than 'formatting' (Madnick & Wang, 1990).

At least in an allusive sense, this archaic usage is still pertinent to the sort of 'composition' we examine here, because *to some extent* all our WCSS designs embody a fixed 'composition' of data visualization elements offered up to the user as something as static as a printed page. The highlighted qualification 'to some extent', however, means that it is neither accurate nor sufficient to equate the entirety of a WCSS central visualization with a printed page. This point will be further elucidated later in the discussion.

Even though some pretty formalized (even algorithmic) structure has been applied to 'information composition', the concept has eluded precise delineation. As such, there's really no clear prior definition of 'information composition' to be used as a precedent or a starting point. Much that is connoted by allusions to 'information composition' has been researched and practiced under other IT labels and other fields such as:

- Database theory / design
- Knowledge representation
- Ontology design
- Hypermedia studies
- Media studies
- Human factors
- Human-computer interaction

The theme of organizing data to suit the needs or accommodate the limitations of a reader or user can be traced back to the earliest considerations of computers as engines for data compilation and distribution. The prospects for flexible data access and presentation envisioned by Vannevar Bush (1945) for his MEMEX concept were a matter of information composition capabilities surpassing those of the print medium. By the late 1960's there was no such thing as a personal computer, but it had already been recognized that users of centralized mainframe-based information systems could benefit from a means for flexibly selecting and composing data. In December 1968 Doug Engelbart and his colleagues gave the first public demonstration of an interactive hyperlinked data display (later developed as AUGMENT). Reitman *et al.* (1969) describe an aid (AUTONOTE) affording the ability to mix and organize data at will.

From that point into the 1980's research into what might be called 'information composition' continued to be framed with respect to large scale or special-purpose systems. The original linkage to database applications continued, and some additional interest was generated in the field of artificial intelligence (where complex visual displays were often associated with expert systems). The rise of personal computers did not motivate much concern for information composition until the arrival of graphical user interfaces such as the Apple Macintosh. By the mid-1980's research and development were increasingly focused on the opportunities such graphical capabilities afforded. The introduction of spreadsheets and desktop publishing capabilities had an indirect effect by giving end users sufficient control over their own data outputs to warrant their becoming concerned about, and knowledgeable on, how best to organize and render their products. With the arrival of rudimentary hypertext applications around this time (e.g., Xerox

PARC's NoteCards; Apple's HyperCard) the organization of data and information became a driving issue in interface design and research (cf. Halasz, 1988).

These general lines of inquiry and development went into a second-stage acceleration with the arrival of HTML and its implementation in the World Wide Web. For the first time, the functionalities previously limited to individual platforms or LAN-based suites of platforms were available to a general - indeed, a universal - audience. As information composition innovations and techniques were carried forward from platform-centric to network-centric deployments, a new wave of applied development ensued. By the close of the 1990's, attention was turning to the means for providing tailored, on-demand information via networks. To accomplish this while drawing on multiple and disparate data resources one had to grapple with issues of 'information composition'.

Ironically, this half-century of technological innovation had the effect of bringing us back around to the very scenario posed at the beginning by Vannevar Bush. He envisioned a desktop facility at which a person could access and manipulate data. At that early point, the mere notion of having such access was a major revelation. By the time the prospect of such universal data access had been achieved, however, much more had been accomplished and learned about how to bring together the raw data generally than about how to configure that data to be really useful to the person sitting at Bush's desk.

A Review of 'Information Composition': Recent Representative Examples

So what is the state of the art in 'information composition'? In this section we shall present three examples of recent research and development work which cite 'information composition' as a construct. These examples will suffice to illustrate some basic points about the state of the art. Based on these points, we shall then discuss what, if anything, must additionally be considered for such information composition approaches to mesh with the sort of 'composition' we perform in designing our WCSS concepts.

Example 1: 'Information Composition' as an Alternative Access Strategy

Lau *et al.* (2002) employ the notion of 'information composition' in describing an adaptive capability for information filtering. These researchers use the construct 'information carriers' to denote data artifacts such as documents, document components, and even individual characters. Their approach to adaptive information retrieval is based on the 'composition' of such information carriers into arbitrary forms as needed. 'Information composition' is therefore addressed as the process of manipulating a hierarchical 'Lego set' of formally-specified 'information carriers' (data artifacts). This leads to an orientation in which one addresses information access as a matter of composing discrete elements into informative products rather than 'filtering' products out of a large mass of data. The authors describe a structured representational format underlying their model and their demonstration application.

This work (and others like it) has the advantage of making information 'composition' amenable to algorithmic processing. This comes at the price of making the application problem statement all about 'data', not 'information'. This approach still leaves the user in the position of having to operate in a 'data-centric' mode (i.e., focused on the background data and the mechanics of processing it, as contrasted with the elements of his / her work domain). Furthermore, this approach is geared to the universal or generic application; it provides no guidance on tailoring access to 'fit' an operational setting. Finally, the scope of concern is circumscribed by the need to simply access data. This approach makes no provision for tailoring data access or presentation with respect to whatever the user may need to do with it once it's supplied.

Example 2: 'Information Composition' in Recombinant Information Spaces

The second illustrative example is drawn from work conducted at the Texas A&M Interface Ecology Lab and reported in Kerne and Sundaram (2003). Playing on the metaphor of recombinant DNA, they are exploring the notion of recombinant information spaces. 'Information composition' in this context becomes a matter of creative (re-) combinations and arrangements of data elements. Unlike the other examples, this work is linked to a specific operational model of user browsing activities, as illustrated in Figure 39.

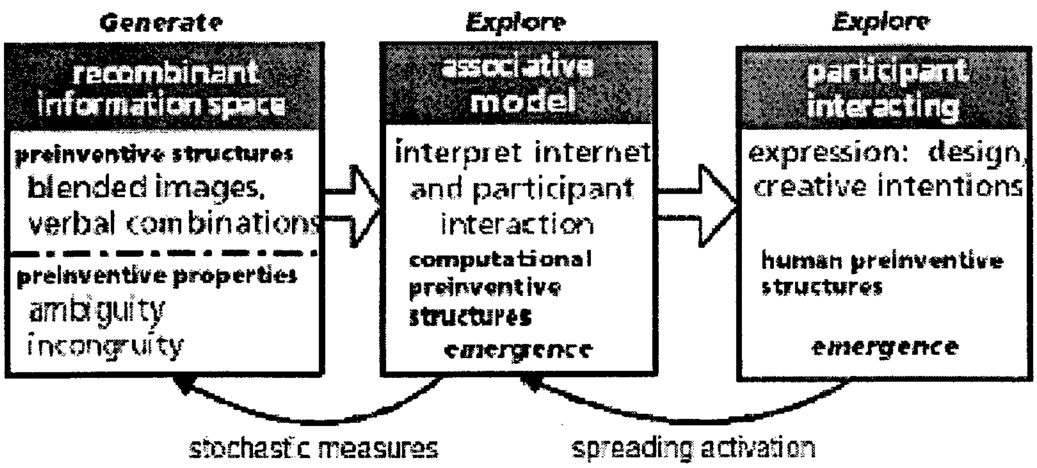


Figure 39: User Browsing Model for Recombinant Information Research

This group developed a structured framework to contextualize data support for exploratory 'recombinant' information activities. This framework was based on decomposition of documents down to the level of atomic data elements and strategies for flexibly blending these atomic data units. In this respect, this example shares the emphasis on modular data elements noted for the 'information carriers' in the first example above.

As was the case for the first example, this recombinant information space work has the advantage of employing a structured data environment which is amenable to algorithmic

processing. One might assume that because this work was predicated on a particular model of user browsing activity it might come closer to being applicable to a 'work-centered' application. However, the only user activity accounted for in this framework is browsing passive display products. There is no real provision for accommodating a user's specific information requirements or derivative options / actions, except to the extent they are managed by the user him-/herself.

Example 3: 'Information Composition' in Tailored Media Products

The third illustrative example comes from the field of media research rather than computer-based IT. Researchers at the Boston University Multimedia Communications Laboratory developed a theoretical model and a demonstration application for 'composing' information (in the form of video and graphic elements) into news segments for television broadcast. In this case, composition was a matter of assembling a sequential news presentation from available data artifacts. Unlike the previous two examples, this research work included attention to the meaningfulness of the product, because it guides its composition with respect to criteria of sequence and narrative coherence. As such, this line of research addressed semantics (ascribed to the end user - i.e., the viewer) to an extent the prior examples did not. Similar to the other two research groups mentioned above, this group developed a structured framework to support their composition strategies. This framework was based on a hierarchy of objects representative of the data artifacts being manipulated, and it was geared to reflect temporal features so as to facilitate sequencing in the material.

As was the case for the prior two examples, this approach has the advantage of involving a structured data model amenable to algorithmic processing. It is also interesting in the sense that it addresses the notion of 'composing' information with respect to time and sequencing - two elements important to TACC operations specifically and to most work activities generally. However, this approach also has the same disadvantages as the other two. It is aimed at passive display products being passively viewed by the end user. There is no real attention to information requirements or derivative options / actions on the part of the viewer.

Discussion

These three examples are sufficient to illustrate the manner in which 'information composition' is treated in current IT research. All have produced useful demonstration products, and all seem to have provided new capabilities for dealing with data overload. However, none of them is directly applicable to improving our WCD information composition activities. The reasons for this claim include:

- *These and other examples of 'information composition' work all tend to focus on 'data', not 'information'.* Our WCD work results in products that provide users with data and the means for generating new data products. However,

our WCSS interfaces are designed with specific regard to the user's *information* requirements, not his / her data traffic.

- *All tend to prioritize 'supply push' (of available data) rather than 'demand pull' (relative to the operational context).* Our WCSS concepts have all been predicated on user control over his / her data displays and the manner in which the support tools are invoked and applied. The examples described above all tend to presume a degree of passivity for the end user. In other words, they tend to address the end user as a 'viewer'. Our target users are not passive viewers of data; they are active manipulators and creators of data products.

- *All give guidance on how one could organize the 'what', but nothing about tailoring it to optimize 'what it means to me'.* By and large, the research and products described above focus on the data itself, not on its meaning (semantics) relative to the intended user. They leave the semantics to the user; they provide the data and let the user make of it what he / she will. Our WCSS projects have tended to focus on the reverse - i.e., focusing on what the user needs to know, so as to determine what data needs to be available.

- *Flexibility is obtained at the cost of high procedural / cognitive burdens.* The first two examples provide the means for users to browse data any way they see fit. However, the procedural and cognitive burdens entailed in managing the browsing process may well counterbalance the payoffs of this flexibility. In any case, our TACC customers are not usually performing such passive free-form browsing in the course of their work activities. As such, the introduction of systems analogous to those described above might have the net effect of inducing more burdens with no practical benefit.

This is not to say that the above-cited examples (and other analogous work) are irrelevant to our WCSS projects. Remember - we are discussing the 'use perspective'. From the 'technical perspective' these sorts of results could be sources of tips and knowledge that could be leveraged to improve (e.g.) TACC data access capabilities.

In the remainder of this chapter we shall turn our attention from the previous 'top-down' attempt to locate useful models and tools to inform our WCSS work to what may be called a 'bottom-up' exploration of information composition as it is evident in our tools and our methodology.

Information Composition and the WCD Process Path

Work-centered design (WCD) is conventionally portrayed as a sequence of four primary steps (cf. Chapter IV, Figure 19):

- Work knowledge capture
- Work-centered requirements analysis
- Work aiding design
- Work-oriented evaluation

This process path is sometimes portrayed in terms of two phases - problem analysis (subsuming the first two steps above) and design synthesis (subsuming the last 2 steps). One means for evaluating the role of information composition in WCD is to determine at which step(s), and in what manner(s), information composition is a matter of concern. A summary assessment is given in Table 37, which lists the points at which information composition is relevant in terms of both the 4-step and the 2-phase frameworks.

Table 37: Information Composition and the WCD Process Path

	PROBLEM ANALYSIS	DESIGN SYNTHESIS
WORK KNOWLEDGE CAPTURE	• Determine the 'composition' of existing information assets and tools. • Collect clues to 'what' and 'what it means to me'.	(Possibly) Initial clues to either info composition features to eliminate or types of features to be desired.
WORK-CENTERED REQUIREMENTS ANALYSIS	• Assess the 'composition' of existing information assets and tools. • Identify gaps, deficiencies, and problems with existing information support. • Project new or modified 'compositions' that would address the problem conditions.	(Possibly) Initial presumptions about either info composition features to eliminate or types of features to be desired.
WORK AIDING DESIGN THE FOCAL STEP FOR PROACTIVE 'INFORMATION COMPOSITION'.		• Specify the information elements and their 'composition' to be embodied in the WCSS. • Define the elements and their organization that optimally meets the need.
WORK-ORIENTED EVALUATION		• Test the new 'composition' for adequacy, usability, and utility. • Modify the 'composition' as required.

Information Composition and Problem / Vantage / Frame

Another construct often invoked to describe the WCD methodology is that of *Problem - Vantage - Frame (PVF).* There are variations on the precise definitions applied to the 3 components of this construct. However, the following basic characterizations are representative:

- *Problem* - A state, condition, or situation being targeted for intervention

- *Vantage* - The optimum or most effective 'point of view' on the subject matter associated with improving the Problem situation.

- *Frame* - An 'instantiated vantage', in the sense of being a specification for implementing the Vantage along with any other features (e.g., controls, options) judged necessary to complete a useful artifact.

The PVF construct was formulated in the wake of the HISA project to describe the orientation to interfaces and their design that had been employed in creating the 'MOG Viewer' (to be discussed later in this chapter). The point was to illustrate an approach which was based on the cognitive capacities of and cognitive demands upon the target user and which prioritized that user's first-person perspective on his / her work as the main context for design creation and evaluation. Effective interfaces must aid the user in recognizing, analyzing and reacting to problems in the course of work. The measure of merit for an interface design is taken to be the degree to which it alleviates burdens or obstacles in resolving such problems. To achieve this end, the design has to provide the user with everything he / she requires to conduct the work at hand with no superfluous, distracting, or error-inducing features.

To determine the best set of interface features, we attempt to fit the interface design to the way(s) in which the worker cognitively addresses problem situations. WCD addresses the subject work in terms of its being an unfolding series of problem solving incidents and / or scenarios. Each such event will be engaged with respect to one or more distinguishable contexts of reference and evaluation. The design of WCSS interfaces is based more on these general contexts than on procedural or functional particulars. This is intended to make the design decision space more tractable, because even though the specifics of a work event may vary endlessly, the set of relevant contexts through which the work is addressed in that event or across multiple such events tends to be stable.

Through work knowledge capture and analysis, we educate ourselves on the work environment and the work as it is practiced. This is the basis for identifying and analyzing the key *Problems* confronted by target users. Next we have to determine the essential features or factors characteristic of the problems to indicate the referential background in which they are best addressed and evaluated. This step requires discovering and describing the optimal point of view for addressing a given problem. This point of view (*Vantage*) is one from which all relevant aspects of the problem are discernible and within which extraneous aspects are de-emphasized or eliminated. The final step is to design interface concepts which afford the user two capabilities - a stable means for addressing the problem from the optimal vantage and a set of affordances for performing the functions necessary to resolving the problem. The specification for a discrete artifact accomplishing these twin objectives is the *Frame*.

Table 38: Information Composition and Problem - Vantage - Frame

PROBLEM	• The Problem specification should provide clues for the requisite new 'composition'. • These clues may include gaps or deficiencies in the current 'composition' in available tools and resources
VANTAGE	• The Vantage sets the fundamental context within which the informative content is to be 'composed' • Since the Vantage is typically the basis for a WCSS' focal visualization, it is in effect a preliminary specification for part of the information composition
FRAME	• Designing the Frame is mainly a matter of 'information composition', to the extent it involves including and organizing features

Table 38 correlates design goals involving information composition (broadly defined) to the three elements of the PVF construct and the design process path they reflect. In the specification of a Problem attention must be paid to the 'composition' of the information assets and support tools the target user currently employs. As one proceeds to formulation of the Vantage and Frame, designers' attention is turned to the 'composition' of an envisioned solution (e.g., a WCSS interface).

Information Composition and the Focus / Periphery Organization

WCSS designs often exhibit a general form consisting of a central visualization affording a vantage on the most pertinent features most pertinent to the work situation at hand. The point is to present data in such a manner as to expedite problem identification and understanding. This theme of centralized focus within a frame design has become a canonical element of our WCSS products.

A single vantage - i.e., a single visualization - may not be enough to permit a user to comprehensively view and assess everything. Sometimes it is necessary to look at a problem situation from multiple angles, each of which may reflect a distinct vantage or variant of a vantage. If the focal data needs to be presented at different levels of abstraction, granularity or detail, the WCSS interface must incorporate controls allowing manipulation of display parameters. When additional data is needed that is not available within the current vantage, the user needs to leave or set aside the current vantage to obtain a different visualization by triggering navigation options. Both these tactics are implemented in WCSS using a single motif, which places the essential visualization 'front and center', then surrounds that visualization with peripheral elements allowing (a) manipulation of the current vantage and / or (b) access to other related vantages.

We call this arrangement of design elements a *focus-periphery organization*. This maintains a capacity for engagement with the entire referential context while focusing the

user's immediate attention on the currently selected vantage and its most crucial features. This minimizes mandatory digressions for the sake of (e.g.) data gathering or interpretation. It therefore reduces the cognitive and procedural burdens placed on the user when navigating and operating within a given vantage.

The focus-periphery organization principle clearly addresses the 'composition' of information on the WCSS interface. As such, it can be considered a specific information composition tactic associated with WCD. In actual practice, 3 distinguishable types of 'composition' can be identified in realizing the focus-periphery organization in a given WCSS concept:

- 'Composing' an effective Focus (the central visualization portraying the required vantage)
- 'Composing' the requisite features for the Periphery
- 'Composing' a best case allocation of features and interplay between Focus and Periphery

An Illustration of the Focus-Periphery Organization

The focus-periphery theme dates back to the first WCSS concept carried forward to development - the 'MOG Viewer' from the HISA project (also sometimes referred to as the 'Port Planner' or 'Port Viewer'). This inaugural WCSS was designed to allow users to evaluate the presence of maximum-on-ground (MOG) conditions for a given airfield (port) within a given 24-hour timeframe. A representative version of this prototype concept drawn from HISA archival documents is illustrated in Figure 40.

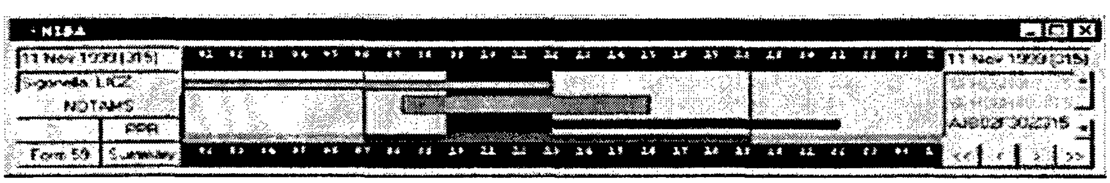


Figure 40: The MOG Viewer from the HISA Project

The MOG Viewer was configured so that the display of flight traffic flow through the given airfield was localized in the center of the interface. To the left and the right of this central visualization were two sets of additional information and control features. On the left (topmost) were text windows providing SA cues on the airfield and timeframe. Beneath these were a set of buttons by which the user could call up additional data (e.g., the Form 59). To the right of the central display are a stacked set of text boxes providing data on the mission displayed. Below these are a set of buttons allowing the user to manipulate the vantage afforded by the central display (e.g., scrolling to examine data for the following day).

Correlating Focus-Periphery Organization with the Concepts of 'Vantage' and 'Frame'

Let us consider how the focus-periphery organization and the Problem-Vantage-Frame approach might be correlated with respect to the structuring of a stereotypical WCSS interface. This is illustrated in Figure 41.

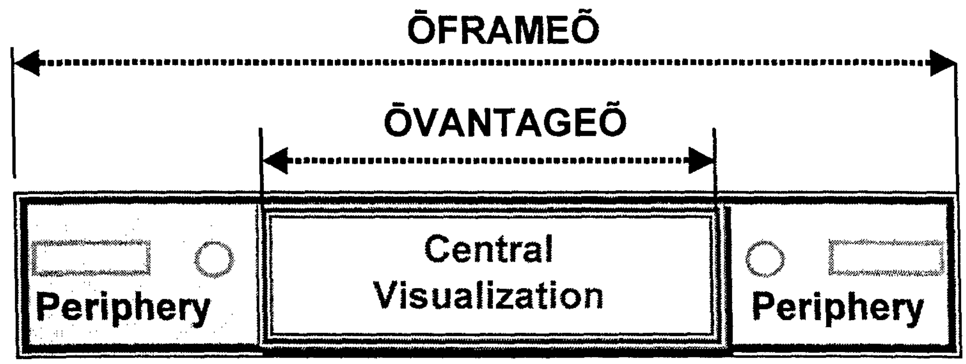


Figure 41: Relationship Between P-V-F and Focus-Periphery in the Interface

The Vantage concept correlates strongly with the intended Focus in a WCSS interface design, and hence with the central visualization. The concept of 'Frame' encompasses the peripheral features augmenting the central visualization (but not the central visualization per se). One can figuratively think in terms of the central visualization being the 'picture' portraying the Vantage and the Frame being the 'picture frame' surrounding it.

Information Composition and the Focus / Periphery Organization

At this point we can start to tie together the WCD themes and principles discussed above and outline a general progression to the WCD process in terms of information composition. In the sections below this progression will be introduced to illustrate the basic visualization design aspects of WCD's Work Aiding Design step.

Correlating the Focus-Periphery Organization with the Design Process

To further demonstrate the integrated nature of our WCD elements, let us consider how the focus-periphery organization and the Problem-Vantage-Frame approach correlate. This is illustrated in Figure 42.

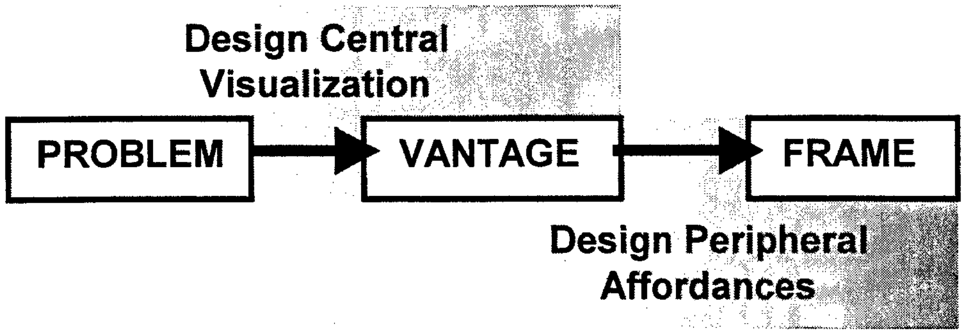


Figure 42: Relationship Between P-V-F and Focus-Periphery in the WCD Process

As Figure 42 indicates, WCSS design typically concentrates first on the specification of the focus (central visualization) in accordance with the specification of the abstract Vantage to be implemented. The process then accretes additional specifications and features to account for an effective periphery, which at least in the abstract is a portion of what must be accounted for in the Frame.

Summarizing the Progression of WCD Visualization Design

To the extent the design of WCSS visualization components can be described as a linear progression of steps, this progression is best described in terms of an accretion of the elements discussed above. An illustration of this basic progression is given in Figure 43. The figure outlines a four-step procedural sequence running from top to bottom.

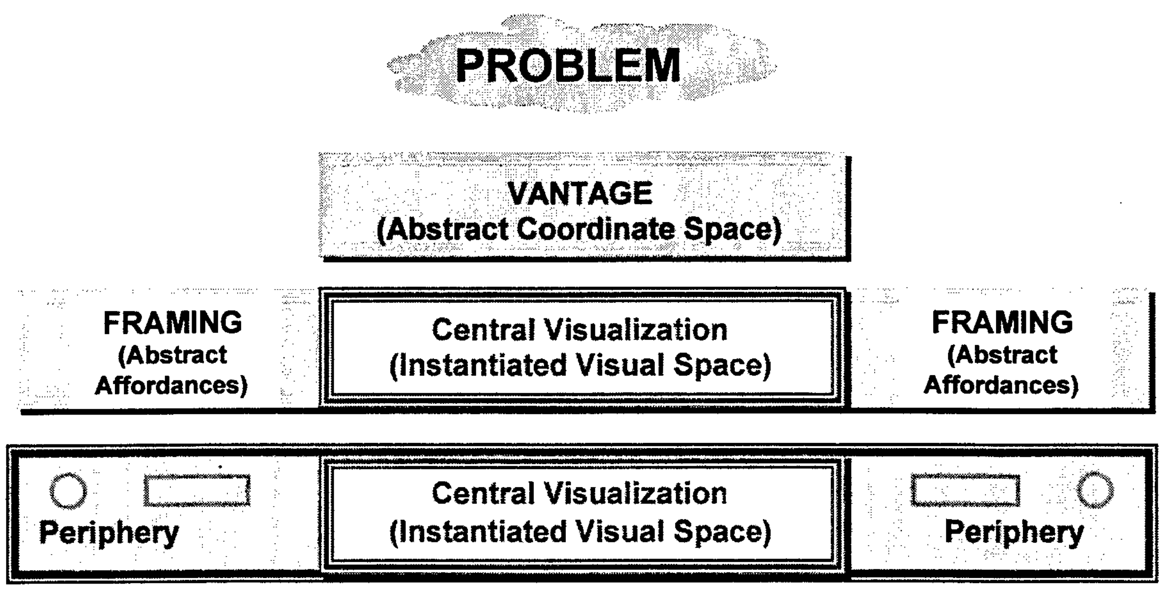


Figure 43: The Basic Progression of WCSS Visualization Design

Given a Problem specification, one must determine the appropriate Vantage for dealing with it. The Vantage is an abstraction connoting the referential matrix or context best suited for portraying and correlating the data so as to optimally inform the user. As such, the Vantage can be construed as a specification for a 'coordinate space' onto or into which the data will be mapped. Once the Vantage specification is stable, the designer(s) can proceed to sketch out a central visualization based on this coordinate space.

At this point (and onward) the designer(s) must also note and allow for any general affordances that must be provided to the user (e.g., for manipulating the central visualization or for invoking external data resources). Because these general affordances are abstract and subject to change at this point, they are referred to as 'framing' in Figure 43. This label is deliberately chosen to insinuate that the features being specified are elements of the Frame. It may take several iterations of design sketching followed by evaluation before a stable set of framing is assembled. The last step is to translate the abstract framing specifications into concrete features of an interface artifact. These constitute the peripheral features of the interface design.

This progression should not be taken as a cookbook recipe that can be simplistically followed step by step. Getting from the initial Problem specification through to a tangible set of central visualization and peripheral features typically involves numerous starts and stops. The transition between any two (or more) steps in this apparent progression might only be finally achieved once several cycles of test and revision are done. Having said that, the outline indicated in Figure 43 is representative of the overall course of design work in our WCSS projects to date.

A Model of Layered Information Composition for Central Visualizations

The most labor-intensive aspect of the WCD design synthesis phase is the generation of specifications for the central visualization intended to translate a designated Vantage into a concrete display. This part of the work is the one most clearly and most deeply intertwined with the notion of 'information composition'. Although the elements to be portrayed on a central visualization are 'data', their collective arrangement and individual features are supposed to constitute 'information' when viewed and grasped by the user.

Over the course of 6 years and 4 projects (HISA, IFM, GAMAT, and WIDE) AFRL's WCSS teams have produced a set of innovative visualization concepts for TACC users. This set is relatively small. However, there are similarities or 'family resemblances' that can be discerned among certain aspects of these concepts and the processes and procedures through which they were created. If there is a specifiable information composition model or protocol that can be considered characteristic of WCSS and WCD, its form (or at least its outline) should be emerging by this point in time. We believe such a characteristic information composition model is evident in the manner in which prior WCSS central visualizations have been conceived and specified.

An exercise was conducted during the final phase of the WIDE 6.2 project in which two specific WCSS concepts were analyzed with respect to the organization of their central visualizations. The two specimens thus studied were the MOG Viewer WCSS concept (from HISA) and the Core element of the timeline tool concept developed during this WIDE project. There were multiple reasons for choosing these particular specimens for analysis, including:

- The similarity in the formats of their central visualization components (both incorporating horizontal 'timeline' representations).
- Their correspondence in the sense of basing their central visualization capabilities on the portrayal of temporally-correlated events
- The fact that both are visually 'rich' in terms of the number of objects portrayed and the scope of denotative and connotative features these objects portray
- The fact that both were clearly designed in (at least general) accordance with the progression and orientations we have come to attribute to WCD specifically

The analysis proceeded by cataloging the objects and types of objects which had been included in the design concept specifications for each specimen. These objects were then correlated with and / or contextualized with respect to the rationale underlying their inclusion in the design specifications. The objects were also categorized in terms of their relative positions, ordering, and features (both dynamic and static) as elements of the visualization. In the end, a set of 8 categories were identified which could be relatively ordered in such a manner as to represent a progression from the most basic visualization criteria to the most specific user functionalities predicated on the holistic set of all other categories' elements.

The set of categories can be construed as corresponding to both (a) a set of layers comprising the logical or conceptual model for a central visualization and (b) a series of relative milestones in the course of creating the visualization specifications. The categories identified in this analysis (presented in order from the most basic / intrinsic to the most derivative / contingent) are as follows:

- *Referential Context* - This category connotes the conceptual or taxonomic context within which all the central visualization's data is to be depicted. The referential context is a general specification for the definitive 'angle' or 'perspective' from which the data is intended to be viewed.
- *Coordinate Space Specifications* - These are the definitions for the referential dimensions onto which data objects are to be plotted. Because our WCSS central visualizations to date have been two-dimensional artifacts, they have

each involved 2 coordinate dimensions. These dimensions may be quantitative or qualitative in nature.

- *Order /Registration Elements* - We cannot provide constructive visualization aiding by simply dumping data into a coordinate space. We need to delineate protocol(s) for organizing and correlating the elements to be displayed. This category of information composition concern addresses such protocols and conventions. Such conventions apply to both quantitative and qualitative dimensions specified in the previous category. If quantitative, they are subject to incremental indexing by numerical value. If qualitative, they are subject to relative ordering with respect to the criteria implicit in the qualitative dimension's specification.

- *Objects* - This category subsumes the discrete visual objects depicting elements of the subject matter. This category circumscribes the basic repertoire of visual items available for the user's inspection and evaluation.

- *Static Object Features* - This category subsumes those persistent object adjuncts or strictly-correlated on-screen features that qualify or enrich a given object's presentation. One example would be a textual label - a separable object whose inclusion is solely for the purpose of fleshing out the portrayal of its associated object of reference. Another example would be any of multiple features strictly applied to indicate attributes of the main object (e.g., visual object size variation to indicate priority or physical size).

- *Dynamic Object Features* - This category subsumes those changeable or mutable object adjuncts or features analogous to those described in the last category. Examples of such dynamic features include width, texturing or coloration of a given object.

- *Meta- / Multi-Object Features* - This category subsumes on-screen objects or phenomena which denote overarching features about, but not of, the displayed objects. A meta-object feature is one which (e.g.) overlays the basic object depiction to connote some additional information about (e.g.) its state or condition. A multi-object feature is one which (e.g.) overlays multiple basic objects so as to connote additional information delineated with respect to all of them as a set rather than each of them individually.

- *On-Display Manipulation Features* - This category subsumed any cues or features made available for manipulating the display as an object of interaction. Such capabilities may pertain to individual objects portrayed within the display, sets of such objects, or features of the display itself (e.g., sizing, ordering, scope of displayed phenomena).

Table 39: General 8-Layer Model for Central Visualization Composition

Visualization Layer	Type of Data / Functions Contained
ON-DISPLAY MANIPULATION FEATURES	Any cues or features for manipulating the display contents
META- / MULTI-OBJECT FEATURES	Overarching features about, but not of, the displayed objects
DYNAMIC OBJECT FEATURES	Mutable object adjuncts (e.g., colors, states) which qualify or enrich the object portrayal
STATIC OBJECT FEATURES	Persistent object adjuncts (e.g., labels) that are always present and associated with a given object
OBJECTS	Discrete visual objects depicting elements of the subject matter
ORDER /REGISTRATION ELEMENTS	Protocol(s) for organizing and correlating displayed objects
COORDINATE SPACE SPEC'S	Referential dimensions on which data objects are to be plotted
REFERENTIAL CONTEXT	The fundamental context within which all data is depicted

This set of 8 categories is summarized in Table 39. The categories listed in Table 39 are organized vertically with regard to their level of generality or scope of determinative effect in the design of a central visualization. The bottom-most category is the most basic and most determinant, while the topmost is the most 'derivative' in the sense that it is defined with respect to all the others. Each of the categories (with the exception of the bottom-most) is conceptually dependent on the one(s) listed below in the sense that the lower-listed categories set the stage for even conceiving its utility and capacities. Each of the categories (with the exception of the topmost) helps to determine and circumscribe the possible form(s), format(s), and function(s) of visualization elements attributed to the categories listed higher in the table.

This research product of the WIDE 6.2 effort is empirical in the sense that it has been derived from demonstrable practices rather than abstract theory. It is relevant because it has been generated with specific regard to AFRL's WCSS and WCD work. These characteristics make the 8-layer model (even in its inaugural form) a more applicable and well-suited information composition schema than the representative approaches and models discussed earlier in this chapter.

The 8-Layer Composition Model and the HISA MOG Viewer

To illustrate the descriptive utility of the 8-layer model, another analytical exercise was undertaken. In this exercise the model was used to descriptively 'deconstruct' the central visualization component of the MOG Viewer which served as the inaugural WCSS concept and prototype. The results of this analysis are summarized in Table 40.

Table 40: The MOG Viewer Deconstructed via the 8-Layer Model

Visualization Layer	Type of Data / Functions Contained
ON-DISPLAY MANIPULATION FEATURES	Ability to grab and drag endpoints of mission on-ground object to simulate modified itinerary
META- / MULTI-OBJECT FEATURES	• Distinguishable visual subregion denoting airfield operating hours • Visual subregion denoting any period of maximum-on-ground (MOG) occurrence
DYNAMIC OBJECT FEATURES	• Color coding of mission objects to indicate general alert condition status • Timepoint designations on time index • Light / dark areas on day / night index
STATIC OBJECT FEATURES	• Mission ID labels • Color coding for organic versus inorganic missions • Bar thickness correlated with aircraft type / size
OBJECTS	• Time indices • Bars representing periods of on-ground presence for any given mission • Day / night index illustrating light / dark periods
ORDER /REGISTRATION ELEMENTS	• Horizontal Axis: • Linear time progression moving from left to right • Extent = 24 hours • Vertical Axis: • Relative ordering of missions based on arrival time • Extent = arbitrary
COORDINATE SPACE SPEC'S	• Horizontal: The timeframe during which mission traffic flows are being visualized • Vertical: Set of mission periods on-ground
REFERENTIAL CONTEXT	Timeframe of operation for a given airfield (port) Generally phrased: TIME and PLACE

The 8-Layer Composition Model and the WIDE Timeline Tool Core Element

A second analytical demonstration of the 8-layer model was performed with respect to the Core component of the timeline tool display concept developed during this project. This exercise was performed with a focus on the Core's central visualization area. The general form of the Core element is illustrated in Figure 44, and a summary of the results of the exercise is illustrated in Table 41.

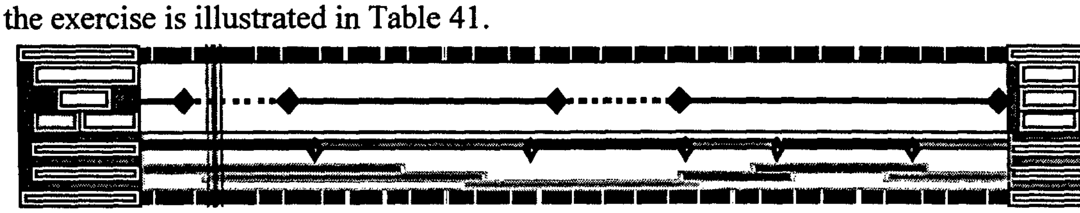


Figure 44: Core Element in the Timeline Tool Design Concept

Table 41: The Timeline Tool 'Core' Deconstructed via the 8-Layer Model

Visualization Layer	Type of Data / Functions Contained
ON-DISPLAY MANIPULATION FEATURES	• Ability to grab and drag endpoints of mission factor object to simulate modified circumstances • Ability to scroll horizontally by manipulating current time indicator
META- / MULTI-OBJECT FEATURES	• Distinguishable visual subregion denoting projected 'windows of opportunity' (e.g., for AR) • Visual subregion denoting on-ground periods at airfields
DYNAMIC OBJECT FEATURES	• Color coding of mission objects to indicate general alert condition status • Horizontal point registration as known or projected schedules change • Position of current time indicator relative to time index • Indicators of early / late status (pending) • Vertical ordering of category depictions - used to denote relative priority for user attention (pending)
STATIC OBJECT FEATURES	• Sortie ID labels • Height (per bar) • Line style (solid versus dotted) • End point delimiters
OBJECTS	• Time indices • Current time indicator • Bars representing periods of applicability or existence for the mission factors and features
ORDER /REGISTRATION ELEMENTS	• Horizontal Axis: • Linear time progression moving from left to right • Extent = Variable from 8 up to 72 hours • Vertical Axis: • Relative ordering of mission factors and elements based on topical categorization • Extent = Delimited by total number of lines dedicated to portraying mission factor categories and individual elements
COORDINATE SPACE SPEC'S	• Horizontal: The timeframe during which mission operations are being visualized • Vertical: Set of mission factors and elements
REFERENTIAL CONTEXT	General timeframe during which mission operations are being conducted Generally phrased: TIME

Summary and Conclusions

WCSS and the WCD methodology rely heavily on effective 'information composition'. It is therefore useful to attempt to contextualize WCSS and WCD in terms of such 'information composition'. It might well be useful to interrelate our own information composition approaches with any other applicable information composition models to be found in the literature. Unfortunately, there are few bodies of work characterized as involving 'information composition'. Of these, all can be readily construed as focusing more on the organization and manipulation of 'data' than on the objective of fusing and presenting such data so as to inform a specific user. Furthermore, none of the representative information composition approaches and models reviewed are work-

centered. Indeed, some of them are not framed in such a manner as to indicate how they could be applied to tailor data and information to the needs of a specific work activity.

In the absence of precedent external corollaries, our WIDE 6.2 information composition research has focused on distilling characteristics of WCD information composition practices and correlating them with our current methodological specifications. With respect to the stereotypical central visualization components of our successful WCSS products, we have conducted an analysis resulting in the specification of an 8-layer model delineating a coherent schema for 'composing' effective work-centered displays. The coherence and consistency of this schema suggests it could serve as a basis for its employment to generate *pro forma* - and conceivably even automated - aiding for the conduct of future WCD efforts.

REFERENCES

Ahanger, G. & Little, T. (1997) A System for Customized News Delivery from Video Archives, Boston University Multimedia Communications Lab Technical Report No. 06-06-1997. Available via WWW: http://hulk.bu.edu/pubs/papers/1997/ahanger-icmcs97/TR-06-06-97node2.html

Alexander, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press.

Alexander, C. (1979). *The Timeless Way of Building*. Oxford University Press.

Alexander, C. (2002). *The Nature of Order*. Oxford University Press.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. & Angel, S. (1977). *A Pattern Language*. Oxford University Press.

Bannon, L. (1991). Shared Information Spaces: Community Technology? Issues in Computer-Supported Work, in *Mutual Uses of Science and Cybernetics: Part 2*, Amsterdam: Thesis Publishers, 23-42.

Bannon, L. & K. Schmidt. (1989), CSCW: Four Characters in Search of a Context, in *EC-CSCW '89: Proceedings of the First European Conference on Computer-Supported Cooperative Work*, 358-372.

Bateson, G. (1987). *Steps to an Ecology of Mind*, Northvale, NJ: Aronson.

Borchers, J. (2001). *A Pattern Approach to Interaction Design*. John Wiley & Sons.

Bush, V. (1945). As we may think, *Atlantic Monthly*, 176 (June), 101-108.

Coplien, J. & D. Schmidt (Eds). (1995), *Pattern Languages of Program Design*. Reading, MA: Addison-Wesley.

De Michelis, G. (1990). *Computer Support for Cooperative Work*. London: Butler Cox Foundation Report, October.

Eggleston, R. G. (2002). Cognitive systems engineering at 20-something: Where do we stand? In M. D. McNeese & M. Vidulich (Eds.), *Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa*. Wright-Patterson Air Force Base, OH: HSIAC Press, 15 - 78.

Eggleston, R. G. (2003). Work-centered design: A cognitive engineering approach to system design. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* Denver, CO: Human Factors and Ergonomics Society, 263-267.

Eggleston, R. G. (2005). *Coursebook: Work-Centered Design (WCD).* AFRL/HECS WCD course training manual.

Eggleston, R.G., Roth, E.M. & Scott, R. (2003). A framework for work-centered product evaluation. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting*, Denver, CO, Oct 13-18, 2003, 503-507.

Eggleston, R.G., & Roth, E.M. (in press). Toward A Framework for Work-Centered System Evaluation.

Eggleston, R. G. & Whitaker, R. D. (2002). Work Centered Support System design: Using organizing frames to reduce work complexity. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting* (Baltimore MD: Sept. 30 - Oct. 4 2002), Santa Monica CA: Human Factors and Ergonomics Society, 265 - 269.

Eggleston, R.G., Young, M.J. & Whitaker, R.D. (2000). Work-Centered Support System technology: A new interface client technology for the battlespace infosphere. *Proceedings of NAECON 2000, Dayton OH, 10-12 October, 2000*, 499-506.

Engelbart, D. (1988a). A Conceptual Framework for the Augmentation of Man's Intellect. In I. Greif (Ed), *Computer Supported Cooperative Work: A Book of Readings*, 35-65.

Engelbart, D. (1988b). Toward High-Performance Knowledge Workers. In I. Greif (Ed), *Computer Supported Cooperative Work: A Book of Readings*, 67-78.

Fincher, S. (2002). Patterns for HCI and Cognitive Dimensions: Two halves of the same story? *Proceedings of the 14th Annual Psychology of Programming Interest Group Workshop*, Brunel University, London, UK.

Gabriel, R. (1996). *Patterns of Software.* Oxford University Press.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Greif, I. (Ed). (1988). *Computer-Supported Cooperative Work: A Book of Readings.* San Mateo CA: Morgan Kaufmann.

Halasz, F. G. (1988). Reflections on NoteCards: Seven issues for the next generation of hypermedia systems, *Communications of the ACM*, 31:7 (July), 836 - 852.

Horn, D., Finholt, T., Birnholtz, J., Motwani, D. & Jayaraman, S. (2004). Six degrees of Jonathan Grudin: A social network analysis of the evolution and impact of CSCW research. *CSCW 2004: Proceedings of the 2004 Conference on Computer Supported Cooperative Work.* New York: ACM Press, 582 - 591.

Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. New York: The Free Press.

Johnson-Lenz, P. & Johnson-Lenz, T. (1992). Groupware: The Process and Impacts of Design Choices. In Kerr, E. & S. Hiltz (Eds). *Computer-Mediated Communication Systems: Status and Evaluation*. New York: Academic Press, 129 - 138.

Kerne, A. & Sundaram, V. (2003). A Recombinant Information Space, Texas A&M Interface Ecology Lab Report. Available via WWW: http://www.andruid.com/papers/recombinantInformationSpace8-10.pdf

Klein, G., Kaempf, G., Wolf, S., Thordsen, M. & Miller, T.E. (1997). Applying decision requirements to user-centered design. *International Journal of Human-Computer Studies*, 46, 1-15.

Klein, G., Ross, K. G., Moon, B. M., Klein, D. E., Hoffman, R. R. & Hollnagel, E. (2003). Macrocognition. *IEEE Intelligent Systems*, 18, 81-85.

Kremer, G. (2005). Understanding an expert's design thinking: A knowledge elicitation approach using the modified teachback method. Working paper for AFRL/HEC, publication pending.

Kuper, S. (2004, February). Global air mobility advanced technologies (GAMAT) ATD final program review. Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Laakso, S. (2001). User Interface Design Patterns. Online library of design patterns. Available via WWW: http://www.cs.helsinki.fi/u/salaakso/patterns/

Lau, R., Hofstede, A. & Bruza, P. (2002). A Study of Belief Revision in the Context of Adaptive Information Filtering. University of Queensland / Queensland University of Technology technical paper. Available via WWW: http://www.fit.qut.edu.au/~terhofst/articles/agent.ps.Z

Madnick, S. E., Siegel, M. & Wang Y. R. (1990). The Composite Information Systems Laboratory (CISL) Project at MIT. Pre-print from *IEEE Data Engineering*, June,1990 issue. Available via WWW: https://dspace.mit.edu/bitstream/1721.1/2305/1/SWP-3157-21957262.pdf

Madnick, S. E. & Wang, Y. R. (1990). Logical Connectivity: Applications, Requirements, and an Architecture. MIT CISL Project Paper #2061-88 (Revision of August, 1988 working paper CIS-90-19). Available via WWW: https://dspace.mit.edu/bitstream/1721.1/2212/1/SWP-2061-19215288.pdf

Mulvehill, A.M. & Whitaker, R. D. (2002). Human Interaction with Software Agents (HISA). Air Force Research Laboratory Technical Report #AFRL-HE-WP-TR-2002-07.

Osga, G. A. (2004). Building task-product-focused user interfaces using information sets and design patterns. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*. Santa Monica CA: Human Factors & Ergonomics Society, 2416-2420.

Pree, W. (1995). *Design Patterns for Object-Oriented Software Development*. Reading MA: Addison-Wesley.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. New York: North-Holland.

Rasmussen, J., Pejtersen, A.M. & Goodstein, L.P. (1994). *Cognitive Systems Engineering*. New York: John Wiley & Sons.

Reitman, W., Roberts, R. B., Sauvain, R. W., Wheeler, D. D. & Linn, W. (1969). AUTONOTE: A personal information storage and retrieval system. *Proceedings of the 1969 24th National Conference of the Association for Computing Machinery*, 67 - 76.

Roth, E. & Scott, R. (2003, March). [Notes from KA sessions with flight planners, mission planners, dips clearance shop and flight managers, Work-Centered Support Technology Development]. Unpublished raw data.

Schraagen, J. M., Chipman, S. F. & Shalin, V. L. (2000). *Cognitive Task Analysis*. Mahwah, NJ: Lawrence Erlbaum.

Scott, R., Roth, E., Malchiodi, E., Deutsch, S.E., Kazmierczak, T., Eggleston, R.G., Kuper, S. & Whitaker, R.D. (2003). Using Software Agents in Designing Work-Centered Support Systems. *IEEE Intelligent Systems* (Special Issue).

Scott, R., Roth, E. M., Deutsch, S. E., Malchiodi, E., Kazmierczak, T., Eggleston, R., Kuper, S. R. & Whitaker, R. (2005). Work-Centered Support Systems: A Human-Centered Approach to Intelligent System Design. *IEEE Intelligent Systems*, 20(2), 73-81.

Stanard, T., Eggleston, R.G., Schmidt, V., Wampler, J.L., Kuper, S. R. & Whitaker, R. D. (2005). User Interface Design Patterns: Defining reusable interface templates for cognitive work. Working paper for AFRL/HEC, publication pending as of March 2005.

Tidwell, J. (2000). COMMON GROUND:A Pattern Language for Human-Computer Interface Design. Available WWW: http://www.mit.edu/~jtidwell/ui_patterns_essay.html

US Army. *Field Manual FM 11-43*. Available via WWW: http://www.globalsecurity.org/military/library/policy/army/fm/11-43/c1s1.htm

Van Welie, M. GUI Design Patterns. Online library of design patterns. Available WWW: http://www.welie.com/patterns/gui/index.html

Van Welie, M., van der Veer, G. C. & Eliens, A. (2002). Patterns as tools for user interface design. In J. Vanderdonckt & C. Farenc (Eds). *Tools for Working with Guidelines Annual Meeting of the Special Interest Group*. Springer-Verlag Telos

Vicente, K. J. (1999). *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer- Based Work.* Mahwah, NJ: Lawrence Erlbaum Associates.

Wampler, J., Whitaker, R., Roth, E., Scott, R., Stilson, M. & Thomas-Meyers, G. (2005). Cognitive work aids for C2 planning: Actionable information to support operational decision making. Paper submitted to the *10th International Command and Control Research and Technology Symposium (ICCRTS),* March 2005.

Whitaker, R. (2003, August). GAMAT work-centered support system (WCSS) review: AMC/TACC ("The Big Picture"). Prepared for AFRL/HECS Cognitive Systems Branch of Air Force Research Laboratory.

Whitaker, R. (2004). WIDE User Interface (UI) Concept Overview / Concept Details. Design analysis and concept presentation to AMC/TACC (in 2 parts), O'Fallon, IL, 6 December 2004.

Woods, D.D. & Christoffersen, K. (2002). Balancing Practice-Centered Research and Design. In M. McNeese & M. A. Vidulich (Eds). *Cognitive Systems Engineering in Military Aviation Domains*. Wright-Patterson AFB, OH: Human Systems Information Analysis Center, 121-136.

Young, M.J., Eggleston, R.G. & Whitaker, R.D. (2000). Direct manipulation interface techniques for interaction with software agents. Paper presented at the *NATO/RTO Symposium on Usability of Information in Battle Management Operations*, Oslo, Norway, April 2000.

Young, M.J. & Eggleston, R.G. (2002). Work-Centered Decision Support. *Proceedings of the NATO/RTO Symposium on Information Management in Command and Control,* Warsaw, Poland, 07-11 Oct, 2002, 15-1 – 15-2.

GLOSSARY

Air Mobility Command - the USAF organizational component responsible for air operations accomplishing airlift and transport objectives. AMC has been the top-level customer for AFRL's WCSS / WCD efforts since 1999.

Alert - any message or cue given a user to connote a condition or state of the subject work to which he / she should appropriately give consideration.

AMC - Acronym for Air Mobility Command

AR - Acronym for aerial refueling.

Artifact - any discrete thing which is generated as a result of deliberate construction - i.e., something that is built, crafted, or produced. To some extent, all artifacts involve a measure of 'design'. The process of 'design' specifies and guides the production of an artifact.

ATC - Air traffic control.

Central Visualization - the WCD principle under which we organize interface displays so as to place the focal visualization of work subject matter in the center of the display palette.

Chartmaker - The label denoting the 'graphics guy' - i.e., the junior level back office staffer responsible for monitoring incoming WX data and generating the forecast charts.

City-Pair - The dyad of departure and arrival airfields used as the basis for generating an ACFP route specification.

Cluster - **1.** An organizing concept employed in the design of the Timeline Tool. A cluster is a structured temporal visualization whose content is delineated with regard to a major category of mission-relevant data. **2.** Any of the subsidiary timeline elements on a Single Mission (Timeline) Display which portrays data pertaining to such a category.

Cognitive Engineering - the field or discipline concerned with applying knowledge of human cognition and cognitive performance to the analysis, design, and / or evaluation of tangible work processes and work artifacts. The label was created by Donald Norman in the early 1980's to connote his vision of an applied research field analogous to the pure research field of cognitive science.

Cognitive Systems Engineering - a particular class of cognitive engineering theory and practice based on the work of Danish engineer Jens Rasmussen.

Cognitive Task Analysis - a term for a structured examination and specification of factors, features, and / or model(s) describing and explaining the cognitive aspects of performance in a given work setting and / or for a given work process.

Cognitive Work Analysis - a variant term which can generally be considered synonymous with cognitive task analysis.

Collaboration - a general term for the process through which two or more workers jointly accomplish a common work objective or produce a mutually-generated work product.

Common Route Definition - A standard protocol for organizing and portraying route data.

Composite (Timeline) Display - An early label for a *Multi-Mission Display.*

Coordination - a general term for the process of correlating, synchronizing, or otherwise organizing multiple workers' activities such that their conduct is coherent as a whole.

Core - the term for a visualization artifact providing a summary overview of temporal data for a given mission. The Core serves as a summary 'meta-cluster' in the individual timeline display and as the entire entry for a given mission when included in a multi-mission display.

CRD - acronym for Common Route Definition

CTA - an acronym for cognitive task analysis.

Cue - any perceptual indicator or tactic employed to provide an interface user with a signal connoting a state or condition relevant to his / her work activity.

CWA - an acronym for cognitive work analysis.

Design Artifact - any artifact generated in the course of, and for the purpose of, a design process.

Design Pattern - a construct originating with architect Chris Alexander in the 1960s, denoting a general form or set of features which is recurrently employed to prescribe a design for a given function or a given situation. Alexander's insight was that certain architectural features (e.g., doors and thresholds) exhibited a high degree of commonality across national, historical, and cultural boundaries. He then began enumerating and categorizing such 'patterns'. In the 1980's and 1990's, the IT research and development community began adopting the notion of design patterns (though in multiple ways and with multiple nuanced variations) and applying it to interface designs. The relevance of the concept of design pattern to WCD is that it is resonant with WCD's emphasis on structural / organizational form as a key design motif. At the extreme, one could say that a highly tailored WCSS is an example of a design pattern of limited generality (relative to Alexander's architectural patterns).

DIP - Acronym for 'diplomatic clearance(s)'.

DIP Summary Palette - A WCSS concept generated during GAMAT Phase II to provide rapid situation awareness on the existing DIP data associated with a given sortie.

Diplomatic clearance - The formal credential according permission from a foreign nation to enter and transit its airspace.

DO - Acronym for 'Duty Officer'.

Drilldown - any process or procedure through which a user 'digs into' a general data set or record to get more detailed data.

Duty Officer - The top-level supervisory position within the Execution Cell at TACC.

Electronic Mission Folder - The name of an envisioned IT application which would provide all units within AMC / TACC with a commonly-accessible structured record for each mission being planned and followed.

EMF - Acronym for Electronic Mission Folder.

Execution Cell - The name for an integrated command and control unit at TACC. The Execution Cell is a physically co-located staff who monitor and evaluate missions starting from 24 hours prior to launch through their actual execution.

FIR - Acronym for Flight Information Region. A FIR is essentially a geospatially-delineated area (region) correlated with a governing authority (e.g., for air traffic control). FIR boundaries generally do not correlate 1-to-1 with political boundaries or other area delimiters.

FIR Boundary - The boundary circumscribing a given Flight Information Region. FIR boundaries are important constructs in flight planning because permissions and authority (e.g., for ATC) begin and end at FIR boundaries.

Flight Planning Palette - A WCSS concept developed during the IFM project (2000 - 2001) and presented to TACC staff in spring 2001. This palette incorporated representations for stepwise FM planning procedure (to provide a 'checklist'), a text subwindow serving as a general work area, and miscellaneous data and alert features. The Flight Planning Palette was conceived as a modular 'clipboard' to be employed in doing flight planning and assembling crew papers. TACC later developed this concept into a tool known as the Sortie Manager.

Flight Visualization Tool - The label used during FY02 and FY03 (Phase I and Phase II, respectively) GAMAT efforts to denote a generic visualization application focusing on flight routing elements. Multiple versions of this tool were presented and discussed

during our GAMAT Phase II presentations. The FVT concept was incorporated in the WCSS deployment concept during our WIDE timeline tool design deliberations.

FM - Acronym for flight manager.

Foreign Clearance Guide - The primary reference resource on diplomatic clearances, and the main reference resource used by the DIP shop. This is available in the form of hardcopy manuals physically kept in the DIP shop work area.

FP - Acronym for flight planner.

FPP - Acronym for Flight Planning Palette.

Frame - the term for a discrete structured depiction of a work domain or a specific aspect of a work domain. A frame can be construed as a concise 'window' on a given aspect of the work subject matter or the work itself. " A structural frame depicts the work field from a specific perspective. In a conventional user interface, structural frames are either only implicitly considered or usually designed based on some logic applied to a set of display elements (widgets). In our approach, organizing frames are explicitly designed and guide the selection and form of display elements that are eventually contained within them." (Eggleston & Whitaker, 2002)

FVT - Acronym for Flight Visualization Tool

G2 - A colloquial shorthand acronym sometimes used around AMC to refer to GDSS-2.

GAMAT - Global Air Mobility Advanced Technologies. GAMAT occurs as both (a) the title of the program under which the work reported here was conducted and (b) an occasional label for the weather visualization applications (GWM-WCSS; WMT) the GAMAT project had developed and demonstrated during FY02 and FY03.

GDSS - Global Decision Support System. The primary repository for mission / sortie information from the point of completion of mission planning through the execution phase.

GDSSII - Global Decision Support System II. The next-generation version of GDSS, currently under development by FSG. Sometimes referred to around AMC as 'G2'.

Graphical User Interface - any composite or unit interface artifact incorporating non-textual, pictorial or iconic elements as its primary mode of visual presentation to the user.

GUI - acronym for 'graphical user interface'.

GWM - an acronym for 'Global Weather Management'. This label was occasionally used to refer to the Weather Management Tool (WMT) artifact prototyped during FY02.

GWM -WCSS- an acronym for 'Global Weather Management - Work Centered Support System'. This label was used to refer to the Weather Management Tool (WMT) artifact prototyped during the FY03 GAMAT effort.

HISA - Human Interaction with Software Agents. This was an AFRL/HES project conducted from 1999 through 2000, aimed at demonstrating the application of intelligent software agent technologies to traditional 'shop-based' AMC/TACC flight planning operations.

IADP - acronym for interface artifact design pattern(s).

ICAO - **1.** International Civil Aeronautics Organization. A regulatory organization overseeing civil aviation issues worldwide. **2.** A shorthand label for the official ICAO code name for a given airfield.

IFM - Integrated Flight Management. This is (a) a formal title for the dispatcher model of flight planning / following being implemented as a component of AMC's Mobility 2000 (M2K) program and (b) a general label for a mode of operations in which TACC staff serve as "dispatchers" and/or "virtual crew members" in supporting air crews.

IMT - Integrated Management Tool. This is a data integration application developed by Federated Software Group (FSG) and deployed as the primary flight planning and following support portal in the "swimming pool" within TACC.

IMT Dashboard - The central interface component of the Integrated Management Tool (IMT), consisting of a large tabular display upon which summary mission data is presented.

Individual Timeline Display - The label originally employed to denote a *Single Mission Display* .

Interface - any artifact through which a user engages, monitors, and / or controls a computer-based information system.

Interface Artifact Design Pattern - a design pattern framed so as to focus on an element or component of a UI presentation or display. A design pattern specification addressing the form or function of what's on-screen.

Interface Usage Design Pattern - a design pattern framed so as to focus on the interaction or engagement between the user and the UI - either this interaction per se or this interaction contextualized with respect to a broader view of the user's work, needs, or requirements. A design pattern specification addressing what the user does with or through the UI.

IT - acronym for 'information technology' - the general label subsuming computer and communications technologies and attendant data and information applications.

IUDP - acronym for interface usage design pattern.

KA - an acronym for *knowledge acquisition.*

Knowledge Acquisition - the process or activity through which researchers (designers, developers, etc.) obtain data and knowledge of the work, work processes, work environment, and workers serving as the focus of their design and development effort. Phrased another way, knowledge acquisition is the process of collecting and analyzing data about the context into which one's outcomes are to be interventions.

Layer - any discrete set of visual data which can be overlaid atop another graphic element in a display.

Layer Controls - the interface elements through which a user may invoke and manipulate data layers.

Lead Time - The amount of time required prior to a stated deadline for processing a DIP (diplomatic clearance) request.

Logbook - A recently-emergent information systems application in AMC / TACC which provides personnel with a mutually-accessible repository for notes, documents, and other data on missions. The Logbook provides a location where textual data on a mission can be accreted and retrieved. The software application affording the Logbook capabilities to TACC is 'DAP'.

MAR - Acronym for Mission Area Representative.

Mission Area Representative - An emerging concept for a TACC task or role which serves as the liaison between planning and execution by monitoring mission 96 hours prior through mission completion. The vision is that Mission Area Representatives are planners that would take over and follow a mission beginning at 24 hours prior to launch. As of FY03 - FY04, the concept of MAR has changed from the WX-specific version we first encountered in our GAMAT FY02 work.

Mission Forecaster - The label denoting the front office WX staffer.

MOG - Acronym for 'maximum on ground' - the term for the maximum number of aircraft that can feasibly be on-ground at a given airfield at a given time.

MOG Viewer - A label sometimes given to the Port Viewer tool (originated in the HISA project) and/or its multiple evolutionary descendants.

Multi-Mission Display - The label for a Timeline Tool visualization which depicts summary information on a set of missions (as contrasted with a single mission). Also referred to in 2004 as a *Composite (Timeline) Display.*

Notice to Airmen (NOTAM) - An announcement issued by an airfield or other authority to notify the aviation community of news, updates, changes, restrictions, etc., concerning access to and operations of a given airfield.

NSN - acronym for National Stock Number, a standardized identifier applied to goods.

Ontology - a structured specification for the most basic and essential elements of reference and meaning in a particular context. As used in WCD, an 'ontology' is a structured specification for the meaningful terms, concepts, and constructs employed by a worker in a given work setting.

OOP - acronym for object oriented programming.

Palette - as used in WCSS and WCD, any discrete on-screen window or similar display element serving as a component of a WCSS. Some WCSS may consist of a single palette; others may be comprised of a suite or set of interrelated palettes.

Peripheral Control - the general WCD principle prescribing that interface elements through which actions / responses are triggered should be arranged peripherally around a central visualization of the work (or work-specific subject matter) at hand.

Port Viewer - an interface concept created in 1999 as part of the HISA project, and dedicated to visualization of relevant phenomena descriptive of operations at a given airfield (port) during a given timeframe. The Port Viewer concept was carried forward as a visualization aid geared to evaluating MOG conditions, so subsequent prototypes were often called 'MOG Viewers'.

Process Path - a specifiable series or sequence of steps describing the essential course or flow by which a process is accomplished.

ROO - Acronym for 'Route Orientation Officer'.

SA - Acronym for 'situation awareness'.

SADP - Acronym for software artifact design pattern.

Senior - A supervisory role with oversight responsibilities during mission execution.

Shift Status Display - The name given the original WCSS concept developed during the IFM project (2000 - 2001) representing a compact highest-level overview over the pending workstream. The basic form was that of a vertically-ordered set of tabs, each of which identified a corresponding mission / sortie, provided minimal ID info on that sorties (arrival / departure ICAO's), and a summary alert indicator to cue the user on that sortie's status. This concept's first prototype implementation was as an auxiliary feature of the GWM-WCSS under the name 'Sortie Palette'.

Single Mission Display - The label for a Timeline Tool visualization for a single individual mission. During the WIDE 6.2 project this application was sometimes referred to as an *Individual Timeline Display* .

SME - an acronym for *subject matter expert* .

Software Artifact Design Pattern -- a design pattern framed so as to focus on an element or component of the software 'behind' or 'beneath' a visible user interface (UI). A design pattern specification addressing the form or function of an information systems application outside the scope of what the user sees and / or directly engages.

Sortie Manager - Name for a flight planning tool originally proposed under the title 'Flight Planning Palette' at the conclusion of the IFM project (Spring 2001). This concept was carried forward by the Flight Managers and developed locally into an actual application.

Sortie Palette - The label for the first interactive demo prototype of the IFM project's concept of a 'Shift Status Display', implemented as an auxiliary feature associated with the GWM-WCSS.

Subject Matter Expert - a representative of the target organization or working population well qualified to serve as an information source on what the work is, how it is conducted, and so forth. SME's are the key points of interaction for the purposes of knowledge acquisition (KA). In WCD we prioritize the people currently performing the target task(s) as SME's.

TACC - Tanker Airlift Control Center - the primary transport flight operations component of Air Mobility Command (AMC).

Timeline Tool - The mission information visualization application developed during FY04 - FY05 in conjunction with the WIDE 6.3 program.

User - anyone who engages and employs a given artifact. For our purposes, we are specifically concerned with users of interactive information systems. A person is defined as a 'user' with respect to the artifact(s) with which he / she engages - presumably in the course of performing tasks and accomplishing work.

Vantage - a term used in WCD to connote the perspective or viewpoint of a given worker with respect to subject matter. Phrased another way, a vantage is a 'point of view' with respect to data engaged in the context of a particular work activity. WCD is largely a matter of determining the optimal set of vantages required to engage work-specific data in a manner conducive to making effective sense of that data.

WCD - acronym for *Work Centered Design.*

WCSS - Acronym for 'work-centered support system'.

Weather Management Tool (WMT) - one of many labels used for the agent-based weather visualization tool this project has developed and demonstrated. This tool has also been referred to as the "GAMAT prototype and the GWM-WCSS".

Work Centered - a term used to connote any design which is predicated on the needs, capacities, and / or limitations of the person expected to employ the artifact being designed in *the context and with respect to the perspective of the work process that person accomplishes via his / her activities and tasks.* This is a more nuanced extension of *user-centered design,* which is geared to accommodating a person in their role as a 'user' (with respect to the artifact itself), and not necessarily in their role as a 'worker' (with respect to what they're trying to accomplish). A good 'user-centered' design facilitates operation of the tool, whereas good 'work-centered' design facilitates employment of that tool in the context of a work process.

Work Centered Design - an approach to interactive information systems design developed by the Human Effectiveness directorate within AFRL (AFRL/HE). WCD builds upon user-centered and participatory design practices, human factors knowledge, and cognitive engineering principles to tailor information systems' interfaces to fit the manner in which workers conduct their actual work processes in an operational environment.

Work Centered Evaluation - the process of evaluating WCSS concepts and designs for the purpose of assessing users' views on their viability, utility, effectiveness, and / or efficiency.

Work Centered Support System - a general descriptor for an interactive information aid or tool which (a) is configured to support rather than supplant the human worker; (b) is designed to optimally ensure situation awareness on the work being supported; and (c) is tailored to serve as a window into the work being performed rather than into the information system itself.

Work Ecology - the concept encompassing the dynamic and interactive milieu within which a worker participates in contributing to an overall work process.

Workflow - a term used to connote the directed 'flow' of tasking and work products in a multi-worker work environment. A workflow describes the directions and options for flowing *any* work through an operational organization, whereas the term 'workstream' refers to the work that is being so directed.

Workstream - any ordered set of items or tasks which a given worker or set of workers must address and / or perform as part of their duties. We use 'workstream' to connote the composite contents of the workload. This makes it distinct from the term 'workflow', which we use to connote the directions and options for directing or passing work items through the organization and its work processes.

WX - Acronym for "weather".

APPENDIX A
Scenario / 'Vignette' Development

This Appendix contains the scenario specifications generated by the WIDE design team during October and November 2004. Our starting point was a set of 'vignettes' received from the AMC organization. These scenarios constituted the candidate set we proposed to utilize in our December 2004 Design Review presentation to our AMC / TACC customers.

The versions presented here represents at least 5 'generations' of revisions on the original set of vignettes. Documentation of the scenario / vignette set revision history and working notes relating to presentation issues (in the December 2004 review) have been removed.

Background: A Multi-Leg Mission

This set of vignettes were framed with regard to an illustrative multi-leg mission spanning multiple days, nations, etc.

Mission ID: TQRJZF300053
Mission Type: CHANNEL
Aircraft: DC008
Tail: N799ALC
Priority: 1B1

The mission itinerary is summarized in the table below.

SORTIE	ETD	ETA	DEP	ARR	PORTS
100	4053:2135	4054:0435	RJTY	WSAP	(Yokota - Singapore)
200	4054:2320	4055:0400	WSAP	FJDG	(Singapore - Diego Garcia)
300	4055:0703	4055:1302	FJDG	OBBI	(Diego Garcia - Bahrain)
400	4055:1525	4055:2115	OBBI	FJDG	(Bahrain - Diego Garcia)
500	4055:2335	4056:0440	FJDG	WSAP	(Diego Garcia - Singapore)
600	4056:0700	4056:1315	WSAP	RJTY	(Singapore - Yokota)

This mission crosses over the following countries: Philippines, Malaysia, Indonesia, Oman, UAE, Qatar. It takes off or lands in the following countries: Japan, Singapore, Bahrain. DIP clearances are needed for all these countries.

<u>Vignette #1</u>

Basic Context: Mission 'Scrub'
Problem Detected: Discrepancy involving planned Theater Slot Time

What's Illustrated: Alert capability; Drilldown to Individual Display; anticipation of problems related to subsequent sortie(s).

Specific Requirements:

- This vignette must be framed with respect to sortie 300 - the only one that involves entering a theater of operations (Bahrain).
- The precise mode of alerting employed to initially cue the user must be specified.

Story Line Summary:

Our system does periodic mission scrubs. Six hours prior to launch time for sortie 300 (i.e., circa day 4055 @ 0103) it detects a problem with Theatre Slot times.

This problem detection occurs while the aircraft is in the middle of sortie 200 (i.e., in flight from Singapore to Diego Garcia).

As such, this vignette illustrates a capability for anticipating problems on a subsequent sortie.

The user acknowledges the alert and employs the timeline tool to examine what the problem is. He does this by drilling down to the Individual Display associated with the given mission / leg. It is at this level of detail that he can discern the alert condition is associated with Theater Slot Time requirements.

Vignette #2

Basic Context: Automated anticipatory alerting; predictive monitoring of status for planned AR rendezvous

Problem Detected: Receiver delay negates possibility of making planned AR rendezvous.

What's Illustrated: Alert capability; anticipatory alerting involving parallel / coordinated mission; drilldown to Individual Display. No diverts or action to resolve problems with current flight will be illustrated.

Specific Requirements:

- The specific mechanism / mode / tactic for alerting recognition of this issue need to be determined.
- The particular representational tactics used to depict waypoints on the flight timeline
- The particular representational tactics used to depict the AR opportunities / problems on the Composite Display

- The particulars of the representational tactics used to depict the associated tanker timeline on the Individual Display

Story Line Summary:

During the course of a mission in flight, air crew position reports are verbally submitted to the TACC. Depending on the location, communications capabilities, etc., these reports may be directly 'phoned in' or they may be passed along from a remote location (e.g., an ATC center) to which they are originally submitted.

NOTE: Portraying the operational circumstances pertaining to verbal position reports and how they get accreted to whatever database (GDSS-2, something local to the timeline tool users) is a tricky business. This edition of the vignette is laid out with respect to an 'intermediate approach' in which someone (unspecified - presumably a flight manager) is manually updating a mission-in-flight record (in an unspecified, and presumably 'local', database) in response to verbal position reports as they arrive.

In this storyboard, multiple such verbal reports arrive at the TACC in relation to the given mission. The reports portray a situation in which the progression of the flight is manifesting an incrementally-increasing degree of delay.

As each of the reported positions / times gets manually accreted to the timeline tool's operant database, there comes a point at which the system detects that the current amount of delay now causes a conflict with accomplishing a planned air refueling rendezvous with a tanker.

The DO sees a general alert on the Composite Display (the timeline for the receiver). This alert is associated with visual evidence on the face of the Composite Display indicating the planned AR rendezvous is now in jeopardy.

He / she drills down on the individual receiver mission to an Individual Display which automatically includes a summary tanker timeline allowing him / her to see both (a) the particulars of the receiver delay underlying the alert and (b) the temporal extent of the window of opportunity for coordinated the two aircraft as intended.

This vignette only goes so far as illustrating the alert and the utility of the timeline display for analyzing the cause of the alert.

Vignette #3

Basic Context: Automated anticipatory alerting; exploitation of data from other sources (in this case WX sources).

Problem Detected: Planned flight route will intersect WX watch area. TACC staff needs to consider how to accommodate weather effects (possibly up to and including rerouting).

What's Illustrated: Alert capability; drilldown to Individual Display.

Specific Requirements:

- The specific mechanism / mode / tactic for alerting recognition of this issue need to be determined.
- The particular representational tactics used to depict intersection of the planned flight with a WX watch area need to be worked out.

Story Line Summary:

The DO gets an alert on a particular mission leg (on the Composite Display). He then drills down to the associated Individual Display to explore what it's about.

As it turns out, a mission leg, if executed as originally planned, turns out to require the aircraft to fly through a weather hazard area (e.g., turbulence or thunderstorms or tropical storms). This may well necessitate rerouting which will result in delays.

Once drilled down to the Individual Display, the DO sees a representation indicating the flight during mission leg X involves intersecting a predicted WX hazard area during a given timeframe.

In this one we illustrate intersection between the flight and a weather watch area. The timeline tool should somehow 'highlight' this intersection (e.g., a 'yellow' to designate that the situation is maybe bad, but subject to decision maker review).

Vignette #4

Basic Context: Incoming communication of problem; maintenance delay; replanning; (2) new problems detected based on replanning.

Problem Detected: Multiple problems on subsequent leg, caused by impact of maintenance delay. Replanning to overcome maintenance delay triggers a new problem, and replanning to accommodate this new problem triggers a second new problem.

What's Illustrated: Alert capability; drilldown to Individual Display; anticipation of problems related to subsequent sortie(s); capability for spawning and manipulating a 'simulation mode'; ability to continue replanning and rechecking to discern and deal with subsequent problems triggered by resolution for an initial problem. Because this vignette involves both drilldown and manipulations on a 'what-if' basis, it is more complex than

basic alerting. In addition, the alerting / dealing with 2 new pop-up problems makes this perhaps the lengthiest of the vignettes to step through.

Specific Requirements:

- The story line for this vignette will involve multiple references to what happens to the DO / user or what he / she does in response.
- To keep the number of 'subsequent sortie glitches' manageable, we need to contextualize this story line with respect to the last or next-to-last leg in the mission.
- The story line requires reference to three views of the timeline tool capabilities:
- Composite Display (starting point)
- Individual Display (as is)
- Individual Display + 'Simulation Mode'

Story Line Summary:

The DO receives a phone call advising him that there will be a maintenance delay now that the aircraft has reached Diego Garcia at the end of sortie 400.

This maintenance delay is anticipated to require an additional 8 hours on-ground time.

This means the next-to-last leg of the mission (Diego Garcia - Singapore) will be delayed 8 hours - leaving Diego Garcia at 4055:0735 and arriving at Singapore 4056:1240.

The user responds to the verbal report of a maintenance delay and employs the timeline tool to examine what the ramifications may be. He does this by drilling down to the Individual Display associated with the given mission / leg.

Because the Individual Display depicts what the system 'knows' (i.e., the state of affairs before the phone call), the user must initiate a 'simulation mode' in which he can manipulate the Individual Display (or a clone thereof) to both (a) check the anticipated new state of affairs and (b) explore any ramifications of that new state of affairs.

The user initiates 'simulation mode'. He manually modifies the state of the display to reflect a delay of 8 hours in leaving Diego Garcia (the simplest modification reflecting the effect of the maintenance delay).

His simulation display 'recomputes' the portrayal of the last 2 legs of the mission (e.g., the new departure / arrival times noted above). The recomputed itinerary for the last 2 legs of the mission is now projected as follows:

SORTIE	ETD	ETA	DEP	ARR	PORTS
500	4056:0735	4056:1240	FJDG	WSAP	(Diego Garcia - Singapore)
600	4056:1500	4056:2115	WSAP	RJTY	(Singapore - Yokota)

New Problem #1

Recomputing sortie 500 in light of a simplistic 8-hour 'slide back' surfaces a problem. A new alert indicator is triggered on the updated simulation mode display. This is associated with an airfield problem.

The airfield problem is that the Singapore airfield (WSAP) is closed during the middle of the day (e.g., for construction). The revised arrival time of 1240 is not feasible. It is not reasonable to try and hurry up the maintenance work to get there any earlier. WSAP will be open for accepting new arrivals at 1600.

The DO (or whoever) decides the resolution of this new problem #1 is to delay takeoff from Diego Garcia for an additional 4 hours. This gives the maintenance people an additional 50% overhead on their projected work time and gets the plane to Singapore some 40 minutes after the airport reopens (hopefully avoiding the initial 'rush' that's certain to occur). He manipulates the 'simulation' to reflect an additional 4-hour delay in taking off from Diego Garcia.

His simulation display 'recomputes' the portrayal of the last 2 legs of the mission (e.g., the new departure / arrival times noted above). The recomputed itinerary for the last 2 legs of the mission is now projected as follows:

SORTIE	ETD	ETA	DEP	ARR	PORTS
500	4056:1135	4056:1640	FJDG	WSAP	(Diego Garcia - Singapore)
600	4056:1900	4057:0115	WSAP	RJTY	(Singapore - Yokota)

New Problem #2

Recomputing sortie 500 in light of an additional 4-hour 'slide back' surfaces yet another problem. A new alert indicator is triggered on the updated simulation mode display. This is associated with a DIP problem.

The DIP problem is that the now-12-hour cumulative delay results in the previously obtained DIP clearance for the Philippines being invalid for the timeframe during which the flight is now projected to overfly that country. The Philippines DIP clearance was originally good until 2000 on day 4056. It now needs to be revised to allow overflight several hours later on the same day as originally planned, as well as to accommodate possible 'spillover' to the following day if the flight is running slow overnight.

Vignette #5

Basic Context: Automated anticipatory alerting; predictive monitoring of status for resource allocated to a subsequent mission.

Problem Detected: Delay in executing one or more legs of the current mission; crew rest requirements; availability of this mission's tail number for its next planned mission.

What's Illustrated: Alert capability; anticipatory alerting involving current scheduling / crew rest constraints / availability of current aircraft for subsequent planned mission; drilldown to Individual Display.

Specific Requirements:

- The specific mechanism / mode / tactic for alerting recognition of this issue need to be determined.
- The particulars of the representational tactics used to depict the crew rest data on the Individual Display.
- The particulars of the representational tactics used to depict the resultant non-availability of the tail number for a subsequent mission to which it's been allocated.

Story Line Summary:

The DO receives a phone call advising him that there will be a maintenance delay now that the aircraft has reached Diego Garcia at the end of sortie 400.

This maintenance delay is anticipated to require an additional 8 hours on-ground time.

This means the next-to-last leg of the mission (Diego Garcia - Singapore) will be delayed 8 hours - leaving Diego Garcia at 4055:0735 and arriving at Singapore 4056:1240.

The user responds to the verbal report of a maintenance delay and employs the timeline tool to examine what the ramifications may be. He does this by drilling down to the Individual Display associated with the given mission / leg.

Because the Individual Display depicts what the system 'knows' (i.e., the state of affairs before the phone call), the user must initiate a 'simulation mode' in which he can manipulate the Individual Display (or a clone thereof) to both (a) check the anticipated new state of affairs and (b) explore any ramifications of that new state of affairs.

The user initiates 'simulation mode'. He manually modifies the state of the display to reflect a delay of 8 hours in leaving Diego Garcia (the simplest modification reflecting the effect of the maintenance delay).

His simulation display 'recomputes' the portrayal of the last 2 legs of the mission (e.g., the new departure / arrival times noted above). The recomputed itinerary for the last 2 legs of the mission is now projected as follows:

SORTIE	ETD	ETA	DEP	ARR	PORTS
500	4056:0735	4056:1240	FJDG	WSAP	(Diego Garcia - Singapore)
600	4056:1500	4056:2115	WSAP	RJTY	(Singapore - Yokota)

New Problem #1

Recomputing sortie 500 in light of a simplistic 8-hour 'slide back' surfaces a problem. A new alert indicator is triggered on the updated simulation mode display. This is associated with a crew duty day problem.

The problem is that the crew's delayed arrival in Singapore at 1240 is late enough to trigger a mandatory crew rest cycle. The crew was 'cutting it close' on the original itinerary, but the new 8-hour delay forces a crew rest in Singapore. There would not have been a crew duty day problem with the original schedule, but there will be with the projected takeoff delay.

The minimum applicable crew rest cycle is 16 hours. This additional 16-hour interval is unavoidable.

The DO (or whoever) invokes the 'simulation mode' on the Individual Display to explore the ramifications of this change in plans. He / she manipulates the 'simulation' to reflect an additional 16-hour delay in taking off from Singapore.

His simulation display 'recomputes' the portrayal of the last leg of the mission (using the new departure / arrival times noted above). The recomputed itinerary for the last leg of the mission is now projected as follows:

SORTIE	ETD	ETA	DEP	ARR	PORTS
600	4057:0700	4057:1315	WSAP	RJTY	(Singapore - Yokota)

New Problem #2

Recomputing sortie 600 in light of an additional 16-hour 'slide back' surfaces yet another problem.

After the DO has manipulated the simulation mode display to reflect the 16-hour delay, he / she triggers a forward-looking automated check for next-subsequent resource commitments. A new alert indicator is triggered on the updated simulation mode display. This is associated with a resource problem.

The resource problem is that the aircraft (tail number) currently in use on the current mission has been scheduled for use in another mission launching from Yokota at 4057:0600. The 16-hour delay on the final leg of the current mission will have the aircraft reaching Yokota over 7 hours after it was supposed to have left on the next mission.

A summary timeline for the affected mission is provided in response to drilldown actions taken on the initially-displayed alert. This illustrates to the DO which mission is affected, and how it's affected.